September 1982

++

ACORN Nieuws

++

De zevende aflevering van ACORN NIEUWS. En alweer dikker dan de vorige. Intussen zijn er weer zo'n 100 leden bijgekomen; we stormen op naar de 400 !

Dat is allemaal wel leuk, maar U weet de consequenties van deze groei. Enerzijds loopt ons het werk uit de hand, anderzijds beginnen we wrevel te bespeuren 'in het veld' in verband met verminderende service. U weet ook het recept: r e g i o n a l i s e r e n . Gelukkig komen de regio's goed van de grond.

Met deze zevende aflevering zouden wij, de initiatiefnemers, Marion Gruijters en Gerard Borghaerts, een grens willen trekken. Wat wij beloofd hebben te doen, n.l. tegen minimale kosten een ACORN COMPUTERCLUB op te richten hebben wij gedaan. Dank zij de royale medewerking is over onze computer een vracht aan gegevens boven water gekomen en voor minder dan de kostprijs verspreid.
GRAAG GEDAAN, maar voor ons is de zaak nu eigenlijk af.

We stellen ons voor de laatste Zaterdag van Oktober alle regionale opperhoofden voor een vergadering bijeen te roepen ter bespreking van het verdere beleid. En voor het samenstellen van een voordracht voor een gekozen bestuur t.b.v. de ALGEMENE LEDENVERGADERING op de laatste Zaterdag van November.

Met deze ACORN NIEUWS is praktisch alles 'de deur uit gewerkt' wat persklaar was, of met de beschikbare mankracht persklaar gemaakt kon worden. De rest zullen we terugsturen naar de REGIO'S. Graag het verzoek om verdere copy persklaar in te sturen, resp. eerst binnen de regio te laten bekijken, uittypen en/of duidelijk uitprinten. Gedeeld werk maakt ook hier licht werk!

++

++

...schijnt in 1982
...keer.

IN DIT NUMMER

2/2e In dit nummer. 3/3a LOSSE BERICHTEN

4. De CMOS Geheugenkaart. Een kleine 200 verstuurd; naar schatting een 140 reeds 'in functie'. Verder eigenlijk weinig te melden.

5. de P E E K O computer. In de handel (Telec) is een ACORNSOFT bandje met deze merkwaardige naam. Het gaat hier om een s i m u - l a t i e programma. Het program simuleert een vereenvoudigde versie van een 6502 en de werking daarvan bij gebruik van de normale machinecode's. Een beter hulpmiddel om in assembler te leren programmeren is mijns inziens nauwelijks denkbaar. Reden om aan Paul Moeys te vragen het bijbehorend Manual te vertalen, wat hij voortreffelijk gedaan heeft. Reden ook, om deze vertaling geheel in Acorn Nieuws op te nemen. Wat zou U ervan denken hiermede regionaal een cursusje te organiseren op Uw bijeenkomsten ?

15. Voor de ATOM is nu ook de taal FORTH beschikbaar. Een versie, vrij van auteursrecht, gaat binnenkort naar het bandjesarchief. Gerard Akkermans nam op zich om een inleiding te schrijven in het gebruik van deze zeer interessante Computertaal. Zo te zien weet Gerard waar hij het over heeft,en dit ook in begrijpelijk Nederlands duidelijk te maken. Bij deze de eerste aflevering.

19. JOTTING schrijft Wim Schreuder en bedoelt tekenen met de lol-stok.

20. Binnenkort worden programma's in de Club niet meer per cassette verstuurd maar door de lucht! Ben Haasen stuurde de gegevens.

21. En per TELEFOON! Een serieële uitgang voor dit doel (en andere) uit de doeken gedaan door Albert Verhey. Zeer duidelijk en zeer volledig ! Werd door veel leden al op gewacht. Vandaar dat deze bijdrage niet naar het Archief gaat maar geheel in Acorn Nieuws.

31. Meerdere leden tekenen met de knuppel; ook André de Bruin. En van het scherm hup naar cassette.

32. Wie muziek uit een electrisch orgel mooi vindt en tranen huilt bij het WILHELMUS uit een computer,wordt aangeraden het foefje te leren op deze bladzijde. Hopelijk vond U het leuk, schrijft Wim.

33. Nog een paar programma's van Gerard Akkermans.

35. We leven in de tijd van Vrouw HOLLE volgens Toon Hermans. Voor enkele leden is zelfs de Acorn Atom nog niet snel genoeg. Arno Millenaar over 'hoe het sneller kan'. Op een verantwoorde wijze!

37. Een BUBBLE sorteerder van Hans Marks.

38. Vele leden hebben nu bovenin hun CMOS-Geheugenkaart de zeer compacte (1k !) Monitor zitten van Roel Heuvel als een onmisbaar stuk gereedschap. Het ding kan erg veel. Bij deze deel 1 van een toelichting op het gebruik van dit gereedschap.

41. Jos Horsmeier maakte meteen maar een hele Toolbox en zond deze toe voor vrij (van auteursrecht) gebruik in de Club. Ter onderscheiding van die van Program Power noemen we dit de Josbox. Beschikbaar voor de Eprom-Programmeerdiensten in de Regio's. Bij deze de handleiding, keurig getypt door Mario Ernst.

45. De 2e aflevering Cursus Machinetaal van Leendert Rijnagte.

47. Regeren is vooruitzien. Coördineren óók. Voorzienbaar is, dat een paar [illegible] de volledige ontwikkeling van de ATOM tot een personal computer, waar werkelijk a l l e s mee kan, in de [illegible]. Het resultaat van dit[illegible] door [illegible] en Arie [illegible] ([illegible]), Roel Heuvel en ondergetekende (hardware) vindt U in een SCHAKELKAART waarvan de prototypes al

gebakken zijn. De kern van de zaak is dat: A. Meer geheugenruimte van de Atom toegankelijk wordt gemaakt (DXXX en EXXX): B. Nieuwe statements logisch kunnen worden ondergebracht: C. Toolboxen e.d. uit de handel ongewijzigd bruikbaar blijven: D. D.m.v. CMOS IC's de Jumpers simpel zijn te onderzoeken en aan te passen aan ieders behoefte en E. Ieder zijn eigen toolboxen kan uitproberen, samenstellen, wijzigen en al dan niet in Eprom zetten.
Kortom: Volledig Baas In Eigen Huis.
U hoeft het allemaal niet meteen te begrijpen. U hoeft de kaart ook niet meteen te bestellen. Kan ook nog niet. Hij wordt eerst in kleine kring beproefd. Als U maar vast weet, dat hij eraan komt.

51. Uit Practical Computing Juni '82 een EPROM-Programmer voor de Atom. Simpeler en goedkoper dan de ZERO-Programmer. Dit tijdschrift heeft overigens vaak goede artikelen m.b.t. de AcornAtom.

53. Hans Schoon over a. de dynamische RAMkaart van Electuur. (Voor wie het nog niet weet: Een dynamische RAM is een erg goedkoop geheugen-IC dat echter zó snel 'vergeet' dat vele malen per seconde dit geheugen opgefrist moet worden. Hiertoe dient een extra 'refreshment' schakeling; deze schakeling geeft vaak geduvel en slurpt stroom. Voor wie een massa geheugen nodig heeft kan het een oplossing zijn. De vroeger populaire dynamische geheugens worden geleidelijk echter achterhaald door de goedkoper wordende statische RAM's. Een 2114 koet hier en daar nog maar ƒ 3.-)

54. Afstandsberekening tussen QTH-Locators. Ook van Hans Schoon.

55. LISP. Tweede deel van een inleiding over deze taal door Jos.

59. MANUAL. Tweede deel van de Nederlandse bewerking door Evert Sanders. Voor geinteresseerden: Zie ook onder LOSSE BERICHTEN Blz.

63. Hard op weg naar het samenstellen van "Ieder Zijn Eigen Toolbox" zijn wij geinteresseerd in kleine, hanteerbare SYSTEEMPROGRAMMA's. Op deze bladzijde 4 gocheme zaken. Zie ook onder LOSSE BERICHTEN.

64. Uit de R E G I O's ! De Regionale Clubs komen van de grond. Totnutoe bericht van een stuk of acht. Zie ook onder losse BERICHTEN.

65. Op weg naar de KNOOP-CEL VOEDING. Een volle AcornAtom op 0,7 Amp !

66. ACORN-USER. Het vóórblad van het eerste nummer van een periodiek van ACORN COMPUTERS LIMITED. Een abonnement gaat £18.- (ƒ 85.-) per jaar kosten. Maar geeft U die centen maar aan uw eigen CLUB zodat we ACORN NIEUWS netjes kunnen laten drukken. Want in ACORN-USERS staat g é é n w o o r d over de ATOM. Volgende maand óók niet. Wie daar iets van begrijpt mag het mij berichten ! Kijk maar:

> See next month's *Acorn User* for:
> ● Telesoftware plans from the BBC ● Review of the BBC micro, Spectrum and Electron ● Using the BBC micro in the office ● Art on the BBC micro ●
> ● The BBC micro goes to primary school ●

Het mag duidelijk zijn, dat de ATOM van het toneel gaat verdwijnen. De machine is commercieel te weinig interessant. Of ik daar rouwig om ben ? Nee.. Eigenlijk niet. Eigenlijk helemaal niet.

Want hoe liggen de zaken in feite : Met het (Soft) kunnen invoeren van elk statement dat we willen en zelfs op verschillende manieren alsmede met het (Hard) openbreken van de structuur van de machine en het kunnen maken en neerzetten van (pseudo)Eproms wáár we maar willen, is de ATOM al geen Atom meer. Wat nog ATOM heet is de [illegible] ROM in socket IC 20; in principe en in [illegible] vervangbaar door andere monitors en interpreters [illegible]

Het is in feite mogelijk om de ATOM om te bouwen tot een verbeterde versie op een paar Eurokaarten en een paar leden zijn daar al mee bezig. Voor mij en voor de zeer vele leden in onze Club, die de Atom gekocht hebben om aldoende te leren hoe een computer in elkaar zit, is dit een grandioos vooruitzicht !

Niet dat we nu alle al zover zijn, dat niet. Een procent of 15 à 20 haspelt nog met het Manual en met pril- tot gevorderd BASIC. Circa 30% komt aardig uit de voeten met Basic maar rommelt nog wat rond in ASSEMBLER. De volgende 30 à 40% verstaat redelijk Assembler en ziet kans om het in één program door elkaar te gebruiken en tenslotte nog zo'n 10- 20% leden, die ik tot de artisten reken. Voor HARDWARE is ook wel zo'n verdeling te maken.

We zijn dus allemaal e r g e n s onderweg, maar het doel is hetzelfde: EEN COMPUTER LEREN KENNEN EN LEREN ZO'n DING NAAR DE HAND TE ZETTEN. En hoe verder je daarmee komt hoe minder interessant het wordt met welke computer je begonnen bent. Het enigste, waar je vrij definitief aan vast zit is de MICROPROCESSOR, in ons geval dus de 6502. We worden (als het goed is) in feite 5502 K E N N E R S.

W A A R H E B B E N W E D I E T E R M M E E R G E H O O R D ?

Juist,je. Bij de KIM-CLUB NEDERLAND. Zo'n jaar of 8 geleden begonnen met 'hun' KIM Computer(tje), in de steek gelaten door de fabrikant, dan aangeleund tegen de Junior en in de steek gelaten door ELECTUUR. Maar de CLUB bestaat nog steeds. Hun Blad heet 'de 6502 Kenner' Dezer dagen had ik een gesprek met de Secretaris van die Club, de Heer R.Uphoff. Wat zij doen is het omgekeerde van wat wij doen. Wij gaan van BASIC uit en gaan dan via Assembler naar machinecode. Zij begonnen indertijd met practisch een blote 6502 en beginnen nu bij hogere programmeertalen uit te komen. Een groeiende groep in hun club begint zich voor BASIC te interesseren. Van Assembler weten ze alles af. Hun club telt (eveneens) ca.400 leden.

U VOELT ZEKER WEL, WAAR IK HEEN WIL. LAAT EENS HOREN HOE U EROVER DENKT !

++

LOSSE BERICHTEN

≠ Regio Noord (Groningen/Friesland/Drenthe) laat weten, dat hun volgende (derde) bijeenkomst is op 14 October. Fens de Vries,Assen.

≠ Paul Smulders, voormalig redacteur van 'Bit voor Bit' in Hobbit, (nu bij een andere baas) schreef een Nederlandse handleiding voor de Acorn Atom. Het werd mij ter beoordeling door de KLUWER DEVENTER toegezonden. Mijn eerste indruk was 'Hoe kun je een handleiding voor gebruik mét een computer nu zó vast inbinden, dat je 2 handen voor het boek nodig hebt! Voor ƒ 3,50 bij van Mameren in Nijmegen dus een ringband erdoor. Wat de inhoud betreft: Tja, nogal wat vragen. Maar laten we niet zeuren en een schaap met 5 poten verwachten. Voor wie 'opgegroeid' is met Bit voor Bit en moeite heeft met Engels: w a r m a a n b e v o l e n. de KLUWER,•ƒ 24,50.

≠ Foppe Bouma is bezig een EDITOR te schrijven waar je U tegen zegt, maar daar mocht ik nog niets van zeggen. Wél, dat hij bezig is om losse routines (READ,DATA,RESTORE,IF..THEN..ELSE,SORTeren?,etc.) in 4k bij elkaar te proppen voor een 4k EPROM,die begint met zijn befaamde JUMPER, waarmee je naar deze en andere gebruikersroutines kunt springen. Als het meezit gaat de Monitor van Roel Heuvel er ook nog in. E.e.a. komt als bandje beschikbaar. Foppe stelde mij voor, een rondvraag te houden inzake speciale wensen bij de leden. Bij deze dan. Maar Uw wens(en) graag per briefkaartje !

≠ Jaap van de Woestijne, U weet wel, kreeg de "747 Flightsimulator" van Bug Byte te pakken en vond het (hoe kan het anders) maar niks. In plaats van een Boeing, die op Heathtrow landen moet, maakt hij nu een DC10 die (uiteraard) op Schiphol landen moet. Het wordt een sterk verbeterd programma en vraagt een 'gestapeld'laag geheugen. Ik houd je eraan, Jaap !

≠ DRUKWERKARCHIEF: Gradus Bouwman berichtte ons, dat de postzegels hem langzamerhand de oren uitkomen. Hij geeft door, dat grotere betalingen ook kunnen op zijn girorekening nr. 1156467,Arnhem. Nu we het tóch over DRUKWERKARCHIEF hebben: Het is natuurlijk het mooist, als een archivaris het materiaal dat hij verzorgt goed kent en z.n. kan toelichten. Het terrein 'Computer' is echter zo breed, dat geen sterveling er alles van kan weten. Bij ons kwam derhalve het voorstel binnen om het drukwerkarchief in meerdere onderwerpen op te splitsen. Te denken valt aan opsplitsen in: DATASHEETS en IC gegevens / LISTINGS en Programma-gegevens / PRINTER en INTERFACE gegevens / TIJDSCHRIFTCOPIEen en UITTREKSELS. Beheerders moeten redelijk netjes en niet te duur ergens kunnen copieeren; een beetje weten waar Abraham de mosterd haalt en zo nu en dan een beetje verzorgd berichtje insturen voor ACORN NIEUWS. Sollicitaties voor één van de onderwerpen worden gaarne ingewacht !

≠ MARJO ERNST, CYPRESSTRAAT 94, 6101 JX te ECHT bood ons hulp aan terzake van de L E D E N A D M I N I S T R A T I E met name voor inlichtingen aan -en inschrijven van- N I E U W E L E D E N

Nu, daar hoeven we niet lang over te denken. Bedankt Marjo.

HET TELEFOONNUMMER (VOOR NIEUWE LEDEN) WORDT NU 04754 - 5653

≠ Verklaringen van niet-commercieel gebruik: wij vestigen er even de aandacht op, dat leden, die ooit deze verklaring instuurden, ook bij verdere bestellingen etc. niet weer opnieuw behoeven te doen. Eén keer is voldoende. Overigens vragen wij al geruime tijd deze verklaringen meteen bij inschrijven als lid.

≠ SCHEMA GEHEUGENKAART. De enigste die het schema van de geheugenkaart zoals deze zijn uiteindelijke vorm gekregen heeft, deskundig kan tekenen en in details toelichten is Hans Spijkerbosch. Hans heeft echter zeer veel tijd gestoken in de ontwikkeling indertijd en heeft er genoeg van. De H.T.S. vraagt ook al zijn tijd. In onderdelen zal het schema in Acorn Nieuws behandeld worden. De decodering staat in A.N. nr. 3 alsmede de pin-aansluitingen. De S.S.U.schakeling in dit A.N. blz.4. De rest volgt successievelijk.

≠ ALGEMENE VERGADERING. Voor de Algemene Ledenvergadering, gepland de laatste Zaterdag van November, zouden wij graag in contact komen met iemand, een beetje in het centrum van het land, die voor deze vergadering een gelegenheid zou willen organiseren. Graag bericht.

≠ Een aantal clubleden heeft n.a.v. een artikel in CHIP de geheugenruimte ≠2000 - ≠27ff d.m.v. 'varkens-stapelen' (pin 14 en 15 IC?) in gebruik genomen. Zonder board buffers geeft dat geen problemen. Mét plaatsing van de boardbuffers t.b.v. de CMOS Geheugenkaart kunnen echter wél problemen rijzen omdat IC5 van het board deze adressen b u i t e n de buffer schakelt. U kunt het probleem oplossen door de (uitgebogen) pin 1 van IC? met 3 diodes te verbinden aan resp pin 7, 14 en 15 van IC6, enwel zódanig gericht dat pin 1 naar massa wordt getrokken door elk van de CS-lijnen.

D E C M O S G E H E U G E N K A A R T

Van de CMOS-Geheugenkaart zijn er nu, 15 september, een kleine 200 (twéé honderd) verstuurd. Naar schatting zijn er nú in de Club circa 140 in gebruik. Ongeveer 25 stuks werden 'gebouwd en getest' besteld en afgeleverd.

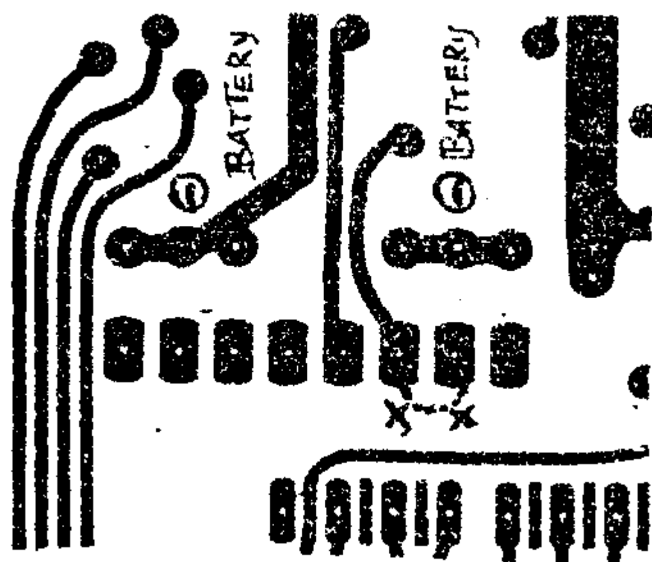
REACTIES kregen wij er, kwantitatief gezien, eigenlijk weinig op. In de lovende zin en 'dik tevreden' een stuk of tien. Dat men het vanzelfsprekend vindt dat de kaart het naar belofte doet, is op zich een compliment. Wij vermoeden ook wel, dat het wel even zal duren tot men in het algemeen de mogelijkheden van de ATOM mét deze kaart, begint door te krijgen. 22k. geheugen aan één stuk waarvan minstens 6k. in BACK-UP ís nogal wat.

Van een stuk of 8 leden kwamen meldingen van niet (direct) functioneren, of liever 'uitval van gestapeld laag geheugen' bij aansluiten van de kaart. Dit werd in alle gevallen verholpen door uit de socket buigen van pootje 1 van IC 5 op het moederboard en verbinden van dit pootje 1(eveneens)aan pootje 7 IC6. Aan dit pootje 7 IC6 zit reeds de $\overline{CS}$ van de gestapelde varkens.

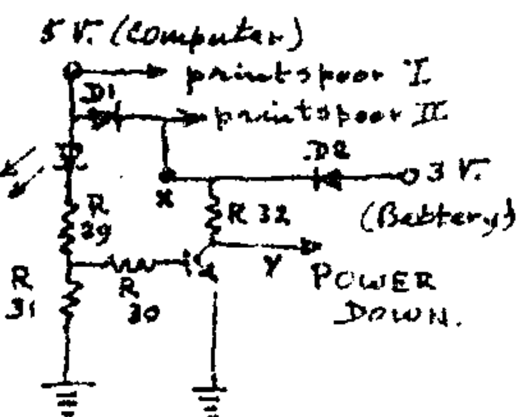
Resterende storingen werden verholpen met doorverbinden van twee soldeereilandjes op de geheugenprint zelf. Enkelen meldden na zelf lang zoeken de storing gevonden te hebben op het moederboard; verkeerde bufferIC's, resp. naar binnen gebogen pootjes.

Onvermijdelijk natuurlijk enkele zelfbouwers, die niet weten te vinden wat - en + van een diode is, ook niet na uitvoerige telefonische uitleg. Danwel onze aanbeveling om glad soldeerwerk te leveren opvolgden door de print langs een slijpsteen te halen.

x DE INGEKOCHTE PRINTEN EN IC's ZIJN NU OP. U KUNT U NOG WEL OP-
x GEVEN VOOR (OOIT) EEN VERDERE SERIE, MAAR WILT U N I E T
x M E E R VOORUIT BETALEN. In ieder geval zullen volgende leve-
x ringen van deze -en volgende- zaken alleen nog plaats vinden via
x de Regionale Opperhoofden. Dit om het werk dat ermee gemoeid is
x een beetje te spreiden. Het was voor ons veel werk,graag gedaan.



Aanbevolen doorverbinding van twee soldeereilandjes op de print wanneer zich een storing voordoet. Moet weer verwijderd worden bij evt. later plaatsen van het bovenkaartje. Een schakelingang vanaf het bovenkaartje wordt met deze doorverbinding zolang aan massa gelegd. Zonder bovenkaartje en zonder deze doorverbinding blijkt deze ingang (soms) te fladderen. Via de ingang stuurt het bovenkaartje straks de buffer.



Bijgaand, op verzoek van enkele klanten, de Batt.Backup schakeling. Bij computer áán moet op punt x minstens 4 Volt staan en op punt y minder dan ½ Volt. Bij computer uit moet op punt x én op punt y minstens 2 Volt staan. De Transistor geleidt dus bij Comp. áán en spert bij Comp. uit. Voor experimenten met Nicad cellen zijn vlakbij op de print 5 losse oogjes ge-etst.

Borghaerts.



de **peeko** computer

1. INLEIDING.

De bedoeling van dit simulatie programma is de "beginner" inzicht te geven in hoe een microprocessor werkt.
Dit programma laat vooral ook zien dat een bepaalde "BYTE" in het geheugen van de processor drie functies kan hebben nl.:

a. Een gecodeerde instructie.
b. Een gedeelte van het adres van een bepaalde geheugenplaats.
c. Gegevens die de processor verwerkt of modificeert tijdens de uitvoering van een programma.

BYTE : Een groep van acht opeenvolgende bits die als één eenheid wordt verwerkt en één geheugenplaats inneemt.

De gesimuleerde processor heeft 50 geheugenplaatsen. Elk van deze geheugenplaatsen kan een enkel decimaal getal bevatten.
De inhoud van het geheugen en van de accumulator is continu zichtbaar, evenzo de toestand van de "carry" en van de "input en output" poorten.

CARRY: Toestandsbit in het toestandsregister van de centrale verwerkings eenheid (CPU) dat aangeeft of het resultaat van een rekenkundige bewerking in de ALU (Schakeling in de processor waarin de rekenkundige en logische beweringen worden uitgevoerd.) groter is dan het grootste getal dat kan worden weergegeven.

De instructieset van de PEEKO is beperkt tot 10 instructies, die veelal overeenkomen met de instructies van de 6502. Maar deze tien instructies kunnen, indien gewenst, veranderd worden.
Het opsporen en verwijderen van fouten (debuggen) wordt aanzienlijk vergemakkelijkt door de mogelijkheid het programma stap voor stap uit te voeren. Hierbij geeft de cursor (program counter) de volgende instructie aan die moet worden uitgevoerd.

PROGRAM COUNTER: Een register, dat het geheugen adres van de volgende instructie die in het computerprogramma uitgevoerd moet worden, bevat.

Als u eenmaal de PEEKO-computer en al zijn mogelijkheden onder de knie hebt is het eenvoudig om echte machinetaal programma's te schrijven voor de 6502.

2. WAT DOEN MICROPROCESSORS.

1. Een microprocessor gebruikt een aantal geheugenplaatsen. Iedere geheugenplaats bevat één byte. Voor de PEEKO-computer is iedere byte één enkel decimaal getal.
2. Elke geheugenplaats heeft een eigen adres. Bij de PEEKO-computer wordt het adres aange geven met twee decimale getallen. De computer leest deze uit van links naar rechts en rij voor rij. Deze adressen zijn bv.: 00,01, 02,............,48,49.
De inhoud van zo'n adres kan bv. voorstellen:
   a. Een instructie. De PEEKO kent 10 verschillende instructies 0 tm. 9.
   b. Een gedeelte van een ander adres. Er zijn twee bytes nodig om een adres aan te geven.
   c. Gegevens die de processor kan verwerken of modificeren tijdens de uitvoering van een programma.
3. De processor zelf heeft registers die gebruikt worden tijdens de uitvoering van een programma. De PEEKO heeft in tegenstelling tot de 6502 maar één register en wel de accumulator (accu).

REGISTER: Een hulpgeheugen of deel van een geheugen dat meestal de capaciteit heeft van een computerwoord en dat een speciale functie heeft in de computer.

4. Enkele speciale adressen kunnen gebruikt worden om informatie uit te wisselen met de "buitenwereld", de zogenaamde input- en outputpoorten. In de PEEKO heeft de inputpoort het adres 98 en de outputpoort het adres 99.

3. PEEKO MONITOR.

Als u het PEEKO programma geladen en "gerund" hebt, dan hebt u toegang tot de PEEKO-monitor.

MONITOR: Programmatuur of apparatuur die de juiste uitvoering van een programma bewaakt en controleert, gewoonlijk d.m.v. stelselmatige checks met een zgn. diagnostic routine die antwoord geeft op vragen omtrent de programmatuur.
Of eenvoudiger: Programma dat de basissysteemcommando's interpreteert en uitvoert.

U kunt de monitorcommando's invoeren middels de hierna volgende (code)-toetsen.

0 - 9 Elke cijfertoets verandert de inhoud van de geheugenplaats die door de cursor wordt aangegeven. De cursor gaat dan naar de volgende geheugenplaats. Bovendien hoort u een piepje dat aangeeft dat de ingevoerde gegevens geaccepteerd zijn.
Van elke ingevoerde instructie wordt ook de mnemonic getoond.(in de linkerbovenhoek van het scherm) Dit eventueel met de daarvoor benodigde argumenten.

G (GO) Laat de PEEKO het programma dat in het geheugen staat uitvoeren, beginnend op de geheugenplaats 0. Dit gebeurt totdat er in het programma een BRK-instructie voorkomt of totdat u de E-toets indrukt (ESCAPE).

E (ESCAPE) Als de PEEKO in de "RUN"-status is stopt het E-commando de uitvoering van het programma en de PEEKO wordt teruggebracht in de READY-status.

SPATIE Door de spatie-toets te gebruiken wordt het programma stap voor stap uitgevoerd en elke uit te voeren insructie wordt in de linker bovenhoek van het scherm getoond.

F (FAST) Hetzelfde als G,alleen wordt het programma sneller uitgevoerd.

← ↕ Controle toetsen om de cursor op iedere willekeurige positie op het scherm te plaatsen.

O (ORIGIN) Plaatst de cursor op geheugenplaats 0.

S (SAVE) Door het S-commando kunt u een PEEKO-programma in het "graphics"-geheugen van de ATOM laden. Dit commando moet gevolgd worden door een decimaal getal van 0 tm. 9 om het programma te identificeren. Als u over niet meer dan 3K "graphics"-geheugen beschikt kunt u alleen de getallen 0 tm. 3 gebruiken.

L (LOAD) Door middel van het L-commando gevolgd door een decimaal getal kunt u een programma uit het geheugen van de ATOM laden in de PEEKO-computer.
(Zo kunt u bv. ook de demonstratie programma's, ADD,COPY,FACTOR eerst laden in het "hoge" geheugen van de ATOM en vervolgens middels dit commando laden in de PEEKO-computer.)

Biedt de mogelijkheid de instructie set van de PEEKO te veranderen.Naast de standaard set van 10 instrc. beschikt de PEEKO over een extra set van 10 instrc. Elke standaard instc. kan vervangen worden door een instrc. uit de extra set.Als de PEEKO in deze mode (I) is kunt u door het "S" kommando teruggaan naar de standaard instrc. set.

P (PRINT) Door dit kommando kunt u het programma dat zich in het geheugen van de PEEKO bevindt via de printer laten afdrukken.
Als u gebruik maakt van een Acorn GP-80 printer gebeurdt dit met extra grote letters.
Voor andere printers is misschien een verandering van de programma regels 9400 tot 9410 noodzakelijk.

N (NEXT) Met dit kommando kunt u een programma regel voor regel "lijsten", waarbij iedere mnemonic op de regel onder "RUN" op het scherm getoond wordt.

> Hiermee kunt u een programma en de toegepaste instrc. set op cassetteband zetten. Dit kommando vraagt eerst om een door u in te voeren filenaam.

< Hiermee kunt u een programma van cassetteband laden. Ook dit kommando vraagt eerst om een door u in te voeren filenaam.

4. INSTRUCTIES

Elke instrc. heeft een "naam" van drie of vier letters, deze "naam" wordt mnemonic genoemd. De mnemonic is doorgaans een afkorting van de beschrijving van wat de instrc. doet.
De verschillende instrc. hebben specifieke gegevens nodig die achter de instrc. moeten worden opgegeven.
Sommige instrc. hebben twee bytes nodig.*) De instrc. "STA" (bewaar de inhoud van de accu) moet bv. gevolgd worden door het adres van de geheugenplaats waarin de informatie moet worden opgeborgen.

```
          2   0 4
          ↑    \↗
 INSTRUCTIE    ADRES
       STA      40
```

BELANGRIJK:
Bij de PEEKO-computer moet de laagste byte van een adres eerst opgegeven worden. De instrc. voorafgaand aan de adrescode interpreteert dit echter als de normale schrijfwijze.
VOORBEELD: Het adres "40" moet aan de PEEKO worden opgegeven als "04".
Sommige instrc. worden "onmiddellijk" gevolgd door een enkele byte, die dan de data voorstelt die door de instrc. moet worden gebruikt.
VOORBEELD: De "laad accu direct met" (LDA @) instrc. moet worden gevolgd door het getal dat moet worden geladen.
Sommige instrc. hebben helemaal geen data nodig.
VOORBEELD: De instrc. "clear carry" zet de waarde 0 in de "carry flag". (De carry flag is een register dat ongeveer het zelfde doet als bv.:"het 1 onthouden bij optel sommetjes".)

5. INSTRUCTIE-SET

De standaard instructie-set van de PEEKO is als volgt:

0 BRK Stop het programma en ga naar de monitor.

*) Om het adres aan te geven van de geheugenplaats waar de instrc.betrekking op heeft.

1	LDA xx	Zet de inhoud van adres xx in de accu.
2	STA xx	Zet de inhoud van de accu in de geheugenplaats met adres xx.
3	CLC	Geef de carry flag de waarde 0.
4	ADC xx	Tel de inhoud van geheugenplaats xx, met inachtneming van de carry, op bij de inhoud van de accu.
5	DEC xx	Verlaag de inhoud van adres xx (met 1).
6	INC xx	Verhoog de inhoud van adres xx (met 1).
7	LDA@x	Laad de accu "onmiddelijk" met het getal (de waarde) x.
8	JNE xx	Spring naar het adres xx als het aan de instrc. voorafgaande resultaat (van een bewerking) niet gelijk is aan nul.
9	JMP xx	Spring naar adres xx.

6. PROGRAMMA'S

Een programma bestaat uit een opeenvolging van instrc. die door de processor een voor een uitgevoerd worden en wel beginnend bij het laagste adres. Dit gebeurt totdat de processor een BRK instrc. tegenkomt.
De "spring" instrc. is ontworpen om deze volgorde te kunnen veranderen en dus de processor zijn volgende instrc. te laten halen van een achter de "spring" instrc. nader gespecificeerd adres.

7. HET PROGRAMMEREN VAN DE PEEKO.

Het hierna volgende laat zien hoe u de instrc. van de PEEKO kunt gebruiken. Door middel van een demonstratie programma zal het gebruik van de meeste instrc. aan u worden duidelijk gemaakt.
Elk programma dat behandeld wordt zal worden opgeschreven als een serie mnemonics, gevolgd door de bijbehorende machienecode en in de meeste gevallen door een tekening van het beeldscherm.

VOORBEELD 1: Het optellen van twee getallen.

plaats	mnemonic	code
00	LDA 40	1 0 4
03	CLC	3
04	ADC 41	4 1 4
07	STA 42	2 2 4
10	BRK	0

Begin op plaats 0,0 en typ de code in. Let erop dat iedere instrc. op het scherm verschijnt onder het woordje "RUN". Typ vervolgens de twee getallen in die moeten worden opgeteld op de plaatsen 40 en 41. Gebruik om op die plaatsen te komen de cursor-controle toetsen.
Als u het programma "stap voor stap" wilt laten "draaien" gebruik dan de spatie toets. Wanneer u nu de spatie toets opnieuw indrukt wordt de eerste instrc.-LDA 40- uitgevoerd. Deze insrc. zet de inhoud van geheugenplaats 40 in de accu.De volgende instrc. (CLC) maakt de inhoud van de carry 0. De ADC instrc. telt de inhoud van adres 41 op bij de inhoud van de accu (met 0 carry) en verandert de inhoud van de accu in het resultaat van de optelling. De STA instrc. zet vervolgens dit resultaat (dus de huidige inhoud van de accu) in geheugenplaats 42.Tenslotte stopt de BRK instrc. het programma.
Als u voor dit optel programma getallen kiest die groter zijn dan 10 dan zal adres 42 alleen het laatste digit bevatten en wordt de carry op 1 gezet. (cyfer)

Als u voorbeeld 1 "gerund" hebt, dan ziet u het hierna volgende schermbeeld:

```
READY   INPUT PORT    OUTPUT PORT
PROG      98 0          99 0
RUN
LDA 40        CARRY 0 ZERO 0 ACC 7

0   1   0   4   3   4   1   4   2   2   4

1   0   0   0   0   0   0   0   0   0   0

2   0   0   0   0   0   0   0   0   0   0

3   0   0   0   0   0   0   0   0   0   0

4   3   4   7   0   0   0   0   0   0   0
    └─+ = ┘
    0   1   2   3   4   5   6   7   8   9
```

VOORBEELD 2. Het optellen van 2 uit 2 digits bestaande getallen.

Dit programma telt de getallen op die staan op de geheugenplaatsen 40,41 en 42,43 en zet het resultaat in de geheugenplaatsen 44,45,46.

plaats	mnemonic	code
00	CLC	3
01	LDA 41	1 1 4
04	ADC 43	4 3 4
07	STA 46	2 6 4
10	LDA 40	1 0 4
13	ADC 42	4 2 4
16	STA 45	2 5 4
19	LDA @ 0	7 0
21	ADC 20	4 0 2
24	STA 44	2 4 4
27	BRK	0

In dit programma wordt een nieuwe instrc. gebruikt. Dit is de LDA@instrc. Deze instrc. "laadt de accu onmiddellijk" met de byte die de instrc. volgt. In dit geval 0.

Het scherm voor voorbeeld 2.

```
READY   INPUT PORT    OUTPUT PORT
PROG      98 0          99 0
RUN
CLC           CARRY 0 ZERO 0 ACC 1

0   3   1   1   4   4   3   4   2   6   4

1   1   0   4   4   2   4   2   5   4   7

2   0   4   0   2   2   4   4   0   0   0

3   0   0   0   0   0   0   0   0   0   0

4   7   6   6   1   1   3   7   0   0   0
    └─    +     =     ┘
    0   1   2   3   4   5   6   7   8   9
```

PROBLEEM 1.

Schrijf een programma met gebruik van de instrc.,LDA,STA,ADC en CLC dat de getallen optelt die staan in de geheugenplaatsen 40, 41 en 42 en het resultaat, dat bestaat uit twee digits, in de geheugenplaatsen 43 en 44 zet.

VOORBEELD 3. Aftel programma.

Het volgende programma trekt van de inhoud van geheugenplaats 40 telkens 1 af. Dit gebeurt in een onafgebroken lus.
:LET OP : 0-1=9 :

plaats	mnemonic	code
00	DEC 40	5 0 4
03	JMP 00	8 0 0

In dit programma komen twee nieuwe instrc. voor namelijk DEC en JMP.
DEC trektéén af van de inhoud van een gespecificeerde geheugenplaats, in dit geval 40. JMP laat het programma verder gaan op een geheugenplaats die achter de instrc. gespecificeerd wordt, in dit geval 00. Deze instrc. komt overeen met de GOTO instrc. in BASIC.
Omdat dit programma een continue lus is, moet E worden ingetypt om uit de "RUN-modus" te ontsnappen.
Door "F" in te typen inplaats van de spatie toets zal de PEEKO het programma uitvoeren zonder te stoppen na elke instrc.

VOORBEELD 4. Tel terug vanaf 99.

Dit programma telt terug naar 0 vanaf een getal dat in de geheugenplaatsen 40 en 41 staat. Bij 0 stopt het programma.

plaats	mnemonic	code
00	DEC 41	5 1 4
03	JNE 00	9 0 0
06	DEC 40	5 0 4
09	JNE 00	9 0 0
12	BRK	0
40		9
41		9

Als u dit programma uitvoert let er dan op dat als de inhoud van geheugenplaats 41 nul wordt, de inhoud van de "zero-flag" 1 wordt.
De inhoud van de zero-flag wordt 1 als de uitkomst van een wiskundige bewerk' nul is. In alle andere gevallen is de inhoud van de zero-flag 0 .

ZERO-FLAG: Een flip-flop, die de logische waarde 1 krijgt indien het resultaat van een instrc. de waarde nul heeft.

De JNE-instrc. doet het zelfde als de JMP-instrc. behalve dan dat de "sprong" veroorzaakt door de JNE-instrc. pas plaats vindt als het resultaat van de voorafgaande bewerking nul is.
In dit programma wordt er ten gevolge van deze instrc. dan ook tot nul geteld.

De INC-instrc. is ongeveer gelijk aan de DEC-instrc., alleen telt deze instrc. 1 op bij de inhoud van het adres dat achter de instc. gespecificeerd is.
Ook deze instr. zet de inhoud van de zero-flag op 1 als het resultaat nul is.
Door de instrc. DEC te vervangen te vervangen door de instrc. INC en de inhoud van de adressen 40 en 41 op nul te zetten, kan het vooraf gaande programma van nul tot 100 tellen.

Op de volgende bladzijde ziet u een tekening van het scherm voor het voorafgaande programma.

READY	INPUT PORT				OUTPUT PORT					
PROG	98 0				99 0					
RUN										
DEC 41			CARRY 0		ZERO 0		ACC 1			
0	5	1	4	9	0	0	5	0	4	9
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	9	9	0	0	0	0	0	0	0	0
	counter									
	0	1	2	3	4	5	6	7	8	9

PROBLEEM 2.

Schrijf een programma dat telt van 99 naar nul en van nul naar 100, dit in een continue lus.

8. NIEUWE INSTRUCTIES.

De instrc. die tot nu toe besproken werden behoorden tot de standaard instrc.-set. Maar er zijn 10 nieuwe instrc.mogelijk die de oude kunnen vervangen. Het I-commando van de PEEKO-monitor vraagt naar het code-getal van de instrc. die vervangen moet worden en naar het code-getal van de nieuwe instrc.. Onder andere door deze mogelijkheid kunt u de PEEKO voor allerlei taken programmeren, dit ondanks de geringe programma omvang.
De werking van de meeste van de nieuwe instrc. is eenvoudig te verklaren omdat deze instrc.vrijwel gelijk zijn aan de instrc. van de standaard set.

SEC zet 1 in de carry.
SBC xx Trek met inachtneming van de carry de inhoud van adres xx af van de inhoud van de accu.
JCC xx Spring naar adres xx als de carry nul is.
JCS xx Spring naar adres xx als de carry 1 is.
JEQ xx Spring naar adres xx als de zero-flag 1 is.
DECA Trek van de inhoud van de accu 1 af.
INCA Tel bij de inhoud van de accu 1 op.
CMP@x Vergelijk de inhoud van de accu direct met x. Als beide gelijk zijn, maak dan de zero-flag 1; zo niet, maak dan de zero-flag 0.
LDA (xx) Laad de accu met de inhoud van het adres dat wordt aangegeven door xx en xx+1
STA (xx) Zet de inhoud van de accu in de geheugenplaatsen met het adres dat wordt aangegeven door xx en xx+1

De laatste twee instrc. zijn indirecte "load- en store" instrc.. Omdat deze instrc. wat ingewikkelder zijn dan de andere instrc. zal hieraan in voorbeeld zes uitgebreid aandacht besteed worden.

De volgende vijf programma's gebruiken sommige van genoemde mogelijkheden. Als u deze instrc. gebruikt moet eerst de instrc.-set veranderd worden. Dit wordt voor ieder voorbeeld apart aangegeven.

VOORBEELD 5. Het aftrekken van twee, twee-cijferige getallen.

De instrc.-set verandert als volgt:
3 SEC
4 SBC

plaats	mnemonic	code
00	LDA 41	1 1 4
03	SEC	3
04	SBC 43	4 3 4
07	STA 45	2 5 4
10	LDA 40	1 0 4
13	SBC 42	4 2 4
16	STA 44	2 4 4
19	BRK	0

```
READY   INPUT PORT     OUTPUT PORT
PROG       98 0            99 0
RUN
LDA 41        CARRY 1 ZERO 0 ACC 8

0  1  1  4  3  4  3  4  2  5  4

1  1  0  4  4  2  4  2  4  4  0

2  0  0  0  0  0  0  0  0  0  0

3  0  0  0  0  0  0  0  0  0  0

4  6  7  1  1  5  6  0  0  0  0
   L     -     -     J
   0  1  2  3  4  5  6  7  8  9
```

Dit programma berekent het resultaat van de opgave:
Trek de inhoud van geheugenplaats 42,43 af van de inhoud van de geheugenplaatsen 40,41 en zet het resultaat in de geheugenplaatsen 44,45.
Als de carry 1 is betekent dat, dat het resultaat niet negatief is. Is de carry nul dan betent dat, dat de inhoud van de geheugenplaatsen 42,43 groter was dan de inhoud van de geheugenplaatsen 40,41 en dus dat de uitkomst negatief is.

PROBLEEM 3.

Schrijf een programma dat twee getallen met elkaar vermenigvuldigt met behulp van de optel-functie.

VOORBEELD 6. Het indirect adresseren.

Verander hiervoor de instrc.-set als volgt:

1 LDA (
2 STA (

Dit programma maakt gebruik van het zg. "indirect adresseren".
Als de inhoud van de geheugenplaatsen 40 en 41 is: 0 2
en de inhoud van geheugenplaats 20 is: 8
dan zal ten gevolge van de instrc. LDA (40) de accu worden geladen met de inhoud van geheugenplaats 20. De inhoud van de accu wordt nu dus 8.
LDA (40) betekent dus:
laad de accu met de inhoud van de geheugenplaats waarvan het adres staat op de geheugenplaatsen (in dit geval): 40 en 40+1.

plaats	mnemonic	code
00	LDA (40)	1 0 4
03	STA (42)	2 2 4
06	BRK	0
40		0
41		3
42		0
43		2

Als u dit korte programma draait zult u zien hoe indirect adresseren gebruikt kan worden om de inhoud van een bepaald adres naar een ander adres te brengen.
In dit geval van adres 30 naar adres 20.

VOORBEELD 7. Het kopieren van een tabel bestaande uit 10 cijfers.

Het programma dat gebruikt wordt voor dit voorbeeld staat ook op cassette en wel onder de file-naam "COPY". Het kan in de PEEKO geladen worden door het commando " " te gebruiken.
Verander de instrc.-set als volgt:
1 LDA (
2 STA (

plaats	mnemonic	code
00	LDA (40)	1 0 4
03	STA (42)	2 2 4
06	INC 40	6 0 4
09	INC 42	6 2 4
12	JNE 00	9 0 0
15	BRK	0
40		0
41		2
42		0
43		3

```
READY   INPUT PORT     OUTPUT PORT
PROG       98 0            99 0
RUN
LDA ( 40 ) CARRY 0 ZERO 0 ACC 8

0  1  0  4  2  2  4  6  0  4  6

1  2  4  9  0  0  0  0  0  0  0

2  1  2  3  4  5  6  7  8  9  0
   ^Copy from here
3  0  0  0  0  0  0  0  0  0  0
   L-----to here
4  0  2  0  3  0  0  0  0  0  0

   0  1  2  3  4  5  6  7  8  9
```

Dit programma laat zien hoe u op eenvoudige wijze,door gebruik te maken van het indirect adresseren, toegang kunt krijgen tot een data tabel. Het programma kopieert een aantal data. Het begin van deze data rij wordt aangegeven op de adressen 40 en 41.
De plaats waarnaar de data gekopieerd moeten worden wordt aangegeven op de adressen 42 en 43. De data staan op de adressen 20 tm. 29 en moeten gekopieerd worden naar de adressen 30 tm. 39.

VOORBEELD 8. Het optellen van twee rijen data.

Ook dit programma staat op cassette onder de file-naam "ADD".

Verander de instrc.-set als volgt:
```
1  LDA (
2  STA (
7  STA
```

Het programma telt twee rijen data, die al in het geheugen zitten, op en zet de uitkomst in het geheugen. Omdat de geheugen ruimte van de PEEKO bep erkt is worden twee rijen data uit het programma zelf opgeteld.

```
plaats   mnemonic    code

 00      CLC         3
 01      LDA (48)    1 8 4
 04      STA 43      7 3 4
 07      LDA (46)    1 6 4
 10      ADC 43      4 3 4
 13      STA (44)    2 4 4
 16      DEC 48      5 8 4
 19      DEC 46      5 6 4
 22      DEC 44      5 4 4
 25      JNE 01      9 1 0
 28      BRK         0

 44                  9
 45                  3
 46                  9
 47                  1
 48                  9
 49                  0
```

```
READY   INPUT PORT      OUTPUT PORT
PROG       98 0            99 0
RUN
CLC           CARRY 0 ZERO 0 ACC 5

0   3   1   8   4   7   3   4   1   6   4
        +   +   +   +   +   +   +   +   +
1   4   3   4   2   4   4   5   8   4   5

2   6   4   5   4   4   9   1   0   0   0
        -   -   -   -   -   -   -   -   -
3   0   5   2   7   1   8   0   0   0   9

4   0   0   0   1   0   3   0   1   0   0

    0   1   2   3   4   5   6   7   8   9
```

Ook dit programma maakt uitgebreid gebruik van indirecte adressering.

VOORBEELD 9. Zoek de factor* van een getal.

Het programma staat ook op cassette onder de file-naam "factor".
Voor dit programma zijn de volgende veranderingen in de instrc.-set nodig:
```
3  SEC
4  SBC
8  JCC
```

(* In dit geval betekent "factor" het grootste gehele)
(deeltal (kleiner dan 10).   )

Het programma berekent de hoogste factor (kleiner dan 10) van een twee cijferig getal. Dit getal staat op de geheugenplaatsen 48 en 49.

```
plaats   mnemonic    code

 00      LDA 48      1 8 4
 03      STA 28      2 8 2
 06      LDA 49      1 9
 09      STA 20      2 0 2
 12      DEC 47      5 7 4
 15      SEC         3
 16      JCC 00      8 0 0
 19      LDA@0       7 0
```

```
plaats   mnemonic    code

 21      SBC 47      4 7 4
 24      STA 20      2 0 2
 27      LDA@0       7 0
 29      SBC 46      4 6 4
 32      STA 28      2 8 2
 35      JNE 16      9 6 1
 38      LDA 20      1 0 2
 41      JNE 16      9 6 1
 44      BRK         0
```

plaats	mnemonic	code
46		0
47		0

```
READY   INPUT PORT    OUTPUT PORT
PROG      98 0           99 0
RUN
LDA 48        CARRY 0 ZERO 0 ACC 5

0  1  8  4  2  8  2  1  9  4  2

1  0  2  5  7  4  3  8  0  0  7

2  0  4  7  4  2  0  2  7  0  4

3  6  4  2  8  2  9  6  1  1  0

4  2  9  6  1  0  0  0  8  9  6
                  factor  number
   0  1  2  3  4  5  6  7  8  9
```

Zet het getal waarvan u de factor wil berekenen in de geheugenplaatsen 48 en 49.

PROBLEEM 4.

Schrijf een programma voor de PEEKO dat een twee cijferig getal deelt door een getal dat uit één cijfer bestaat. U hebt daarvoor in ieder geval de instrc. SBC en SEC nodig en waarschijnlijk ook nog wel enkele andere. Gebruik hierbij het FACTOR-programma als voorbeeld.

9. INVOER- EN UITVOER POORTEN.

INPUT: Het "voer" van een centrale verwerkings eenheid dat via een toetsenbord of een extern geheugen wordt overgebracht naar het interne geheugen.

OUTPUT De uitvoer van data naar randapparatuur of een extern geheugen.

Deze twee poorten, die gebruikt kunnen worden via de geheugenplaatsen 98 en 99 van de PEEKO-computer, komen overeen met de A- en B-poort van het 6522 VIA IC. Dit IC kan in de ATOM geplaatst worden.
de ingangspoort "leest" in principe 4 bits van poort A in en zet deze op geheugenplaats 98 van de PEEKO.
Als de "invoer" groter is dan 10 dan wordt deze teruggebracht tot één cijfer en wordt de carry-flag op één gezet.
De uitgaande poort "99", gebruikt de VIA-poort B.
Omdat een eventueel aangesloten printer ook het 6522 IC gebruikt, moet u alle andere aan dit IC aangesloten apparatuur afkoppelen en de printer aansluiten als u een afdruk van het programma op papier wil hebben.

10. ANTWOORDEN OP PROBLEMEN.

PROBLEEM 1.

plaats	mnemonic	code
00	CLC	3
01	LDA 44	1 0 4
04	ADC 41	4 1 4
07	STA 44	2 4 4
10	LDA@0	7 0
12	ADC 11	4 1 1
15	STA 43	2 3 4
18	LDA 44	1 4 4
21	ADC 42	4 2 4
24	STA 44	2 4 4
27	LDA 43	1 3 4
30	ADC 11	4 1 1
33	STA 43	2 3 4
36	BRK	0

PROBLEEM2.

plaats	mnemonic	code
00	DEC 41	5 1 4
03	JNE 00	9 0 0
06	DEC 40	5 0 4
09	JNE 00	9 0 0
12	INC 41	6 1 4
15	JNE 12	9 2 1
18	INC 40	6 0 4
21	JNE 12	9 2 1
24	JMP 00	8 0 0

1.WAT IS FORTH?

FORTH IS EEN PROGRAMMEERTAAL, DIE ONGEVEER 13 JAAR GELEDEN DOOR C.H. MOORE IS UITGEVONDEN. DE TAAL IS ZEER GESCHIKT VOOR HET BESTUREN VAN APPARATUUR M.B.V. EEN COMPUTER. DE MEESTE HOGERE PROGRAMMEERTALEN ZIJN DAARVOOR TE TRAAG EN HET SCHRIJVEN VAN MACHINETAAL ROUTINES IS LASTIG.
ENKELE EIGENSCHAPPEN VAN FORTH ZIJN:
- FORTH PROGRAMMA'S HEBBEN EEN HOGE EXECUTIE-SNELHEID.
- FORTH HEEFT WEINIG GEHEUGENRUIMTE NODIG.
- FORTH IS UITBREIDBAAR.
- FORTH IS INTERAKTIEF.
- FORTH IS EEN GESTRUCTUREERDE TAAL.

OP AL DEZE EIGENSCHAPPEN KOMEN WE NOG TERUG.

2.DE STACK

VOOR DE BEREKENINGEN GEBRUIKT FORTH EEN STACK (DE PARAMETER-STACK). DIT BETEKENT:
- ALLE GETALLEN DIE WORDEN INGETIKT, KOMEN BOVEN OP DE STACK.
- ALLE OPERATIES HALEN GETALLEN VAN DE STACK EN ZETTEN HET RESULTAAT VAN DE BEWERKING WEER OP DE STACK.

WE ZULLEN EEN VOORBEELD BEKIJKEN.

WE TIKKEN IN: 2 4 6 3

DE GETALLEN 2,4,6,3 STAAN NU OP DE STACK, DE 3 BOVENOP EN DE 2 ONDEROP.

VERVOLGENS TIKKEN WE: +

NU WORDEN DE 6 EN DE 3 VAN DE STACK GEHAALD EN DE SOM (6+3=) 9 WEER OP DE STACK GEZET. DEZE BEVAT NU DUS DE GETALLEN 2,4,9.

TYPEN WE NU: MAX

DAN WORDEN 4 EN 9 VAN DE STACK GEHAALD EN HUN MAXIMUM (=9) WORDT TERUGGEZET. DE STACK BEVAT NU DE GETALLEN 2,9.

WE TYPEN NU: -

DE GETALLEN 2 EN 9 WORDEN VAN DE STACK GEHAALD EN HUN VERSCHIL (2-9=) -7 WORDT WEER TERUGGEZET.

ALS WE NU INTYPEN: .

DAN WORDT HET BOVENSTE ELEMENT (-7) VAN DE STACK GEHAALD EN AFGEDRUKT. DE STACK IS NU LEEG.

KORTOM, HET UITREKENEN EN AFDRUKKEN VAN 2-MAX(4,6+3) GAAT IN FORTH MET: 2 4 6 3 + MAX - . (RET)
OP HET BEELDSCHERM ZAL DAN TE ZIEN ZIJN:

```
2 4 6 3 + MAX - . -7 OK
```

ALS DE OPDRACHT OP DE JUISTE MANIER IS UITGEVOERD REAGEERT FORTH ALTIJD MET "OK".

VAN DIT VOORBEELD KUNNEN WE VERSCHILLENDE DINGEN LEREN:

- DE STACK WERKT VOLGENS HET "LAST-IN FIRST-OUT" PRINCIPE. DIT BETEKENT DAT HET GETAL, DAT HET LAATST OP DE STACK IS GEZET, ER ALS EERSTE WEER WORDT AFGEHAALD. WE KUNNEN DIT VERGELIJKEN MET EEN STAPEL BORDEN!!

- DE SPATIE IS IN FORTH EEN SCHEIDINGSTEKEN. DE VERSCHILLENDE GETALLEN WORDEN ALLEEN DOOR EEN SPATIE GESCHEIDEN.

- ACORN-FORTH WERKT ALLEEN MET INTEGER GETALLEN. ER BESTAAN FORTH-VERSIES DIE OOK MET "FLOATING-POINT" GETALLEN KUNNEN WERKEN. ACORN-FORTH KENT ZOWEL ENKELE- ALS DUBBELE-PRECISIE INTEGERS. OP HET VERSCHIL KOMEN WE NOG TERUG; VOORLOPIG BEPERKEN WIJ ONS TOT DE ENKELE-PRECISIE GETALLEN. DIT ZIJN 16-BITS GETALLEN, ZODAT DE WAARDE DIE ZE KUNNEN AANNEMEN LIGT TUSSEN -32768 EN 32767 (OFWEL TUSSEN 0 EN 65535, INDIEN WE ZE BESCHOUWEN ALS GETALLEN ZONDER TEKEN).

- FORTH IS OP DE MANIER ZOALS WE HEBBEN VOORGEDAAN TE GEBRUIKEN ALSOF HET EEN REKENMACHINE IS. FORTH GEBRUIKT EEN POST-FIX NOTATIE, DIE DE GEBRUIKERS VAN H.P. REKENMACHINES BEKEND ZAL VOORKOMEN.

WE ZULLEN EEN AANTAL STACK-OPERATIES OP EEN RIJTJE ZETTEN:

+	BEREKENT DE SOM VAN DE BOVENSTE TWEE ELEMENTEN.
-	BEREKENT HET VERSCHIL VAN DE BOVENSTE TWEE ELEMENTEN.
*	BEREKENT HET PRODUKT VAN DE BOVENSTE TWEE ELEMENTEN.
/	BEREKENT HET QUOTIENT VAN DE BOVENSTE TWEE ELEMENTEN.
MOD	BEREKENT DE REST VAN DE DELING VAN DE BOVENSTE TWEE ELEMENTEN.
MAX	NEEMT DE GROOTSTE VAN DE BOVENSTE TWEE ELEMENTEN.
MIN	NEEMT DE KLEINSTE VAN DE BOVENSTE TWEE ELEMENTEN.
ABS	BEREKENT DE ABSOLUTE WAARDE VAN HET BOVENSTE GETAL.
MINUS	DRAAIT HET TEKEN VAN HET BOVENSTE ELEMENT OM.
1+	TELT 1 OP BIJ HET BOVENSTE ELEMENT.
2+	TELT 2 OP BIJ HET BOVENSTE ELEMENT.
2*	VERMENIGVULDIGT HET BOVENSTE ELEMENT MET 2. DIT IS SNELLER DAN: 2 *

DIT WAREN ENKELE REKENKUNDIGE BEWERKINGEN, MAAR ER ZIJN OOK ANDERE BEWERKINGEN OP DE STACK MOGELIJK:

DUP	DUPLICEERT HET BOVENSTE ELEMENT. 1 2 DUP GEEFT: 1 2 2
DROP	VERWIJDERT HET BOVENSTE ELEMENT. 1 2 3 DROP GEEFT: 1 2
SWAP	VERWISSELT DE BOVENSTE TWEE ELEMENTEN. 1 2 SWAP GEEFT: 2 1
OVER	COPIEERT HET 2E ELEMENT VAN DE STACK BOVENOP DE STACK. 6 4 3 2 OVER GEEFT: 6 4 3 2 3
ROT	VERWIJDERT HET DERDE ELEMENT EN PLAATST HET BOVEN OP DE STACK. 4 5 6 7 8 ROT GEEFT: 4 5 7 8 6
I PICK	COPIEERT HET I-DE ELEMENT VAN DE STACK NAAR DE TOP VAN DE STACK. 1 2 3 4 5 6 4 PICK GEEFT: 1 2 3 4 5 6 3
I ROLL	VERWIJDERT HET I-DE ELEMENT EN PLAATST HET BOVEN OP DE STACK. 1 2 3 4 5 6 4 ROLL GEEFT: 1 2 4 5 6 3

WE ZIEN NU DAT:
1 PICK IS EQUIVALENT AAN DUP
2 PICK IS EQUIVALENT AAN OVER
2 ROLL IS EQUIVALENT AAN SWAP
3 ROLL IS EQUIVALENT AAN ROT

FORTH KENT OOK LOGISCHE OPERATOREN:

AND BEREKENT DE BIT-VOOR-BIT LOGISCHE AND VAN DE TWEE BOVENSTE ELEMENTEN. 15 21 AND GEEFT: 5
OR BEREKENT DE BIT-VOOR-BIT LOGISCHE OF VAN DE TWEE BOVENSTE ELEMENTEN. 15 21 OR GEEFT: 31
XOR BEREKENT DE BIT-VOOR-BIT LOGISCHE EXCLUSIEVE-OF VAN DE BOVENSTE TWEE ELEMENTEN. 15 21 XOR GEEFT: 26

3. INPUT EN OUTPUT IN FORTH

FORTH KENT DE VOLGENDE INPUT/OUTPUT ROUTINES:

. HAALT HET BOVENSTE GETAL VAN DE STACK EN DRUKT HET AF.
EMIT HAALT HET BOVENSTE GETAL VAN DE STACK EN DRUKT HET AF ALS ASCII CHARACTER. 64 DUP EMIT . GEEFT: @ 64 OK
KEY LEEST EEN CHARACTER IN VAN HET TOETSENBORD EN ZET HET ALS GETAL OP DE STACK.
." DRUKT EEN TEKST AF. DE TEKST MOET WORDEN AFGESLOTEN MET " ." HALLO" GEEFT: HALLO
CR SPRING NAAR HET BEGIN VAN DE VOLGENDE REGEL.

INWENDIG REKENT ACORN-FORTH ALTIJD BINAIR. DE INPUT EN OUTPUT KAN ECHTER GESCHIEDEN IN ELK GEWENST TALSTELSEL. STANDAARD ZIJN AANWEZIG DE "DECIMALE" EN DE "HEXA-DECIMALE" BASIS. OMSCHAKELEN GEBEURT VIA DE COMMANDO'S: "HEX" EN "DECIMAL". HOE JE KAN OVERGAAN OP EEN ANDER TALSTELSEL WORDT NOG BEHANDELD.

4. HET PROGRAMMEREN IN FORTH

PROGRAMMEREN IN FORTH IS EIGENLIJK NIETS ANDERS DAN HET TOEVOEGEN AAN DE STANDAARD-FORTH FUNCTIES VAN JE EIGEN FUNCTIES. ALS JE VINDT DAT EEN FUNCTIE ONTBREEKT, DAN MAAK JE HEM GEWOON ZELF! DEZE ZELF-GEMAAKTE FUNCTIES ZIJN DAN OP DEZELFDE MANIER TE GEBRUIKEN ALS DE STANDAARD-FORTH FUNCTIES.

WE DEMONSTREREN DIT WEER AAN DE HAND VAN EEN VOORBEELD

WE TYPEN IN: : X↑2 DUP * ;

WAT HEBBEN WE NU EIGENLIJK GEDAAN:
: GEEFT AAN DAT WE EEN NIEUWE FUNCTIE GAAN DEFINIEREN.
X↑2 DIT IS DE NAAM VAN ONZE NIEUWE OPERATOR. DEZE WORDT ALTIJD METEEN NA DE : GEZET.
DUP * DIT IS WAT DE NIEUWE OPERATOR MOET DOEN. IN DIT GEVAL MOET DUS HET KWADRAAT VAN HET BOVENSTE GETAL OP DE STACK WORDEN BEREKEND.
; HIERMEE WORDT DE DEFINITIE AFGESLOTEN.

TYPEN WE NU VERVOLGENS: 5 X↑2 .

DAN ZAL DE COMPUTER ANTWOORDEN MET: 25 OK.

WE ZIEN DAT DE NIEUWE FUNCTIE OP DEZELFDE MANIER GEBRUIKT WORDT ALS DE STANDAARD-FORTH FUNCTIES. HIJ KAN DUS OOK IN ALLE DAARNA VOLGENDE DEFINITIES GEBRUIKT WORDEN. BIJVOORBEELD:

```
: X↑4 X↑2 X↑2 ;
```

3 X↑4 . GEEFT NU: 81 OK

IN ACORN-FORTH MAG EEN NAAM VAN EEN FUNCTIE MAXIMAAL 31 KARAKTERS LANG ZIJN. EEN NAAM KAN NOOIT SPATIES BEVATTEN.

EEN FORTH PROGRAMMA BESTAAT UIT EEN RIJ ZELF-GEMAAKTE FUNCTIES, DIE STEEDS VERDER GECOMBINEERD WORDEN. HET GEHELE FORTH PROGRAMMA WORDT UITEINDELIJK GEREPRESENTEERD DOOR EEN ENKEL WOORD.
OM EEN PROGRAMMA IN FORTH TE SCHRIJVEN, MOET DE TAAK VAN HET PROGRAMMA IN STEEDS KLEINERE TAAKJES WORDEN VERDEELD. UITEINDELIJK BLIJVEN DAN ZEER KLEINE TAKEN OVER, DIE MAKKELIJK IN FORTH KUNNEN WORDEN GESCHREVEN. DEZE WORDEN DAN GECOMBINEERD TOT EEN HEEL COMPUTER-PROGRAMMA. WIJ ZULLEN HIERVAN NOG VOORBEELDEN TE ZIEN KRIJGEN.

WE ZULLEN NOG EEN OPERATIE DEFINIEREN.

```
: - 3 * ;
```

WE HEBBEN NU DE OPERATIE "-" GEDEFINIEERD ALS "VERMENIGVULDIG MET 3". WE ZIEN DUS DAT VERSCHILLENDE FUNCTIES DEZELFDE NAAM MOGEN HEBBEN. ALLEEN DE LAATSTE DEFINITIE VAN EEN FUNCTIE KUNNEN WE NU GEBRUIKEN, DUS:

ALS WE INTYPEN: 5 3 - .

DAN ZAL DE COMPUTER REAGEREN MET: 9 OK

WE KUNNEN NU DUS NIET MEER TWEE GETALLEN VAN ELKAAR AFTREKKEN. DAAROM IS ER EEN FUNCTIE IN FORTH, DIE ZORGT DAT WE DEZE DEFENITIE ONGEDAAN KUNNEN MAKEN:

FORGET X↑4 — FORTH IS NU "X↑4" EN ALLE LATERE DEFINITIES VERGETEN. DE STANDAARD-FORTH FUNCTIES ZIJN BESCHERMD! FORGET DROP LEVERT EEN FOUTMELDING OP.

ALS WE WILLEN WETEN WELKE DEFINITIES WE TER BESCHIKKING HEBBEN DAN KUNNEN WE GEBRUIK MAKEN VAN:

VLIST — WE KRIJGEN DAN EEN LIJST VAN ALLE ZELF-GEMAAKTE EN ALLE STANDAARD-FORTH DEFINITIES.

DE VOLGENDE KEER ZULLEN WE HET HEBBEN OVER: VARIABELEN, DO-LOOPS, IF-ELSE-THEN CONSTRUCTIES, BEGIN-UNTIL, BEGIN-WHILE-REPEAT, EN WE ZULLEN EEN VOORBEELD PROGRAMMAATJE LATEN ZIEN.

J O T T I N G

Dit programma heb ik geschreven om, d.m.v. het ons allen bekende "lol-stokje" , tekeningen op het beeldscherm te kunnen maken.
Iedere keer wordt er in ≠BØØ1 gekeken wat daar voor een getal staat normaal is dat 255 (≠FF).
Bekijken we de volgende regel : IF ?≠BØØ1=247;X=X+1
Als wij nu het knuppeltje naar rechts drukken dan wordt er "1" bij X opgeteld.
Dat geld ook zo voor de andere richtingen.Schuin plotten doet ie ook.
Stel dat wij nu echter iets "ontplotten" willen omdat, laten we zeggen, U een tekenfout heeft gemaakt: Wel dat is heel erg een voudig, U drukt gewoon,als toegift op het b.v.naar rechts drukken van het knuppeltje, ook nog even het drukknopje in.
Gaat het plotten U te snel, geen nood de Plotvertragingsfaktor helpt u uit de brand. Vul daar b.v. maar eens 1000 in en u kunt rustig plotten.
Uitleg genoeg nu ? Hier volgt de listing :

```
  1 REM W.A.SCHREUDER
  2 REM VOSSENLAAN 16
  3 REM 9751 GE HAREN(GN)
  4 REM JOTTING-13-8-82
  5 P.$12
 1Ø IN."GRAFISCHE MODE"L
 15 IN."WAAR WILT U BEGINNEN TE PLOTTEN?....X"X"Y"Y
 17 IN."PLOTVERTRAGINGSFAKTOR"S
 18 P."OM TE STOPPEN,DRUK shift"'
 19 F.B=Ø TO 60;WAIT;N.B
 2Ø CLEAR L
 25 PLOT13,X,Y
 7Ø DO
 73 F.B=Ø to S;N.B                    to is natuurlijk TO
 75 K=≠BØØ1
 8Ø IF?K=239Y=Y+1;PLOT13,X,Y                (tussen 9 en Y moet ;)
 85 IF?K=251;Y=Y-1;PLOT13,X,Y
 9Ø IF?K=253;X=X-1;PLOT13,X,Y
 95 IF?K=247;X=X+1;PLOT13,X,Y
1ØØ IF?K=237;X=X-1;Y=Y+1;PLOT13,X,Y
1Ø5 IF?K=249;X=X-1;Y=Y-1;PLOT13,X,Y
11Ø IF?K=231;X=X+1;Y=Y+1;PLOT13,X,Y
115 IF?K=243;X=X+1;Y=Y-1;PLOT13,X,Y
12Ø IF?K=238;Y=Y+1;PLOT15,X,Y
125 IF?K=25Ø;Y=Y-1;PLOT15,X,Y
13Ø IF?K=252;X=X-1;PLOT15,X,Y
135 IF?K=246;X=X+1;PLOT15,X,Y
14Ø IF?K=236;X=X-1;Y=Y+1;PLOT15,X,Y
145 IF?K=248;X=X-1;Y=Y-1;PLOT15,X,Y
15Ø IF?K=23Ø;X=X+1;Y=Y+1;PLOT15,X,Y
155 IF?K=242;X=X+1;Y=Y-1;PLOT15,X,Y
16Ø UNTIL ?≠BØØ1=127
165 P.$12;END
```

........................VEEL PLEZIER....................................
P.S. Op dit moment ben ik bezig alle programma's uit het Magic boek om te bouwen tot Joysticking en op een band te zetten.
Als het klaar is zal ik het bandje aan u toesturen.

Hoogachten Wim Schreuder

HET UITZENDEN VAN COMPUTERPROGRAMMA'S

Beste clubgenoten / zend- en luisteramateurs.

Sinds kort is het mogelijk om aan de radiocontroledienst van de PTT schriftelijk toestemming te vragen tot het uitzenden van digitaal gemoduleerde signalen.
Voor de zendamateurs onder onze clubleden betekent dit dus het uitzenden van o.a. computerprogramma's.
Ook de luisteramateurs kunnen de uitgezonden programma's opnemen en op deze wijze een extra dimensie aan hun hobby toevoegen.
In tegenstelling tot hetgeen oorspronkelijk werd verwacht, hoeven de programma's niet in Basicode te worden uitgezonden, maar kunnen de diverse typen computers in hun eigen "dialekt" met elkaar communiceren.
Toestemming tot bovengenoemde experimenten kan door A, B en C gelicenseerden SCHRIFTELIJK worden aangevraagd bij de;
RADIOCONTROLEDIENST
POSTBUS 570
9700 AN GRONINGEN

Toestemming wordt door de RCD SCHRIFTELIJK verleend onder toevoeging van onderstaande aanvullende bepalingen:

a. Als maximale bandbreedte en klasse van uitzendingen mag uitsluitend 4KØØA1D, 1ØKØA2D, 4KØØF1D en 16KØF2D worden toegepast;
b. De uitzendingen mogen uitsluitend in de frequentieband 144 - 146 MHz en 430 - 440 MHz plaatsvinden;
c. De maximale tijdsduur van elke uitzending bedraagt 5 minuten;
d. Bij het begin en bij het einde van elke uitzending dient de roepnaam van het amateurstation tenminste tweemaal in spraak (conform art.7) of morsetekens, gemoduleerd volgens de in dat gedeelte van de frequentieband gebruikelijke klasse van uitzending, te worden uitgezonden. Tevens dient voor de aanvang van de uitzending te worden aangekondigd dat een begin wordt gemaakt met het uitzenden van digitaal gemoduleerde radiosignalen;
e. Indien ambtenaren van de radiocontroledienst dit noodzakelijk achten dient u hen in de gelegenheid te stellen (conform het bepaalde in art. 13) na te gaan of het uitgezonden digitaal bericht verband houdt met het bepaalde in art.6, lid 1 van de aan uw zendmachtiging verbonden machtigingsvoorwaarden;
f. De toestemming wordt slechts eenmalig verleend en wel voor de duur van één jaar, ingaande op de datum van deze brief.

P.S. Vermeld bij uw aanvraag: merk computer, type microprocessor en de baudrate.

TOT WERKENS. PE1FTX Ben uit Merkelbeek.

RS 232-C interface voor de ACORN ATOM

Albert Verhey
Isabellaland 99
Den Haag
070-477000

Inleiding:

Om data te verzenden van een computer naar een randapparaat of andere computer zyn er twee verschillende methoden mogelijk. Dit zijn parallelle overdracht en seriële overdracht. Bij parallelle overdracht worden voor een byte data 8 verschillende lijnen gebruikt als ingang of uitgang van de computer.
Bij seriële overdracht is dit slechts 1 enkele lijn.

De ATOM beschikt al over de mogelijkheid om data parallel te verzenden met behulp van het V.I.A. i.c. (6522). Om een veelheid aan bedrading tebeperken kan men seriële overdracht toepassen. In zijn meest eenvoudige vorm is dit slechts een 3-draads verbinding van computer met randapparaat.

Om de data van de databus (parallel) om te zetten in seriële data kan men een enkele poort van de V.I.A. gebruiken, echter er is dan een vrij groot programma nodig wat dan ook nogal gecomplicoerd van opbouw is. Voorts krijgt men problemen bij het transporteren van data in "full duplex" (zenden én ontvangen over een enkele - verbinding op hetzelfde moment). Een betere methode is om deze functies te laten verzorgen door een enkel i.c. In de 65XX serie is er een dergelijke chip aanwezig. Dat is de 6551 ACIA (Asynchronous Communications Interface Adapter). Dit i.c. noemt men ook wel een UART, zoals de 8251 van INTEL die dezelfde mogelijkheden heeft voor computers met 8080 of 8085 processor.

De 6551 bevat alle benodigde electronica om data van parallel naar serie om te zetten volgens een programmeerbare codering, en vice versa. Ook kan de i.c. serie data verzenden en ontvangen op 16 verschillende snelheden (baudrates).
Worden alle zend- en ontvangsignalen van deze ACIA nog omgezet in +12V/-12V niveaus heeft men een RS 232-C interface. RS 232-C is een Amerikaanse normalisatie voor seriële transmissie van computers met allerlei randapparatuur, de Europese equivalent is de V24 norm. Het verschil tussen beide is de connector die men toepast, RS 232-C gebruikt een 25-pol. D-connector met genormaliseerde aansluiting en de V24 schrijft helemaal geen bepaald type connector voor.

De Schakeling:

De hieronder beschreven schakeling bevat een 6551 met interne baudrate-generator, een bijbehorend kristal, voor alle i/o lijnen optionele inverters (via stropjes te selecteren), lijn-drivers en -receivers om ervoor te zorgen dat de signalen op RS 232-C niveau gebracht worden. Alle verbindingen van de ACIA met de PL8 bus van de ATOM zijn voorzien van Pull-up weerstanden om verrassingen te voorkomen. Ongebruikte inverters zijn via weerstanden met de 5V voeding verbonden, de voedingsspanning van de verschillende i.c.'s zijn ontkoppeld bij de i.c.'s en de +12V/-12V voeding is nog eens extra afgevlakt teneinde zo zuiver mogelijke voedingsspanningen te verkrijgen.

Voorts vindt U in het schema enkele i.s.'s die tot doel hebben om het niveau van de ingangen en uitgangen van de ACIA op het genormaliseerde spanningsniveau van RS 232 te brengen.(+12 Volt voor een "0" en -12 Volt voor een "1")
De voeding maakt de schakeling compleet. Deze hoeft slechts de beide i.c.'s 2 en 3 te voeden.
De complete hardware staat beschreven in een artikel dat inmiddels in het drukwerkarchief is opgenomen. Om alle mogelijkheden van deze interface op te sommen is Acorn Nieuws te klein,dus zal ik slechts de voornaamste vermelden.

De interface kan geprogrammeerd worden voor de volgende functies:
baudrate(50,75,110,135,150,300,600,1200,1800,2400,3600,4800,7200,9600 en 19200 baud)
aantal stopbits(1,1½,2 stopbits)
data woordlengte(5,6,7,8,bits overdracht)
echo-toestand,interrupt aan/uit,ontvanger aan/uit,zender aan/uit,reset.

Bij de in het drukwerkarchief aanwezige pakket documentatie is ook een programmavoorbeeld dat de ACIA kan sturen gevoegd.

Omdat de omvang van dit project nogal groot is om in 1 keer te beschrijven zal ik in de volgende Acorn nieuws het een en ander vertellen over hoe men met deze interface kan werken.

Tot slot dan nog wat over de aansluiting op de ATOM. Er wordt gebruik gemaakt van de nog vrije connector PL8. Deze connector bevat de adreslijnen voor het gebied ≠B400-≠B7FF. De interface bestijkt slechts de adressen ≠B400-≠B403.

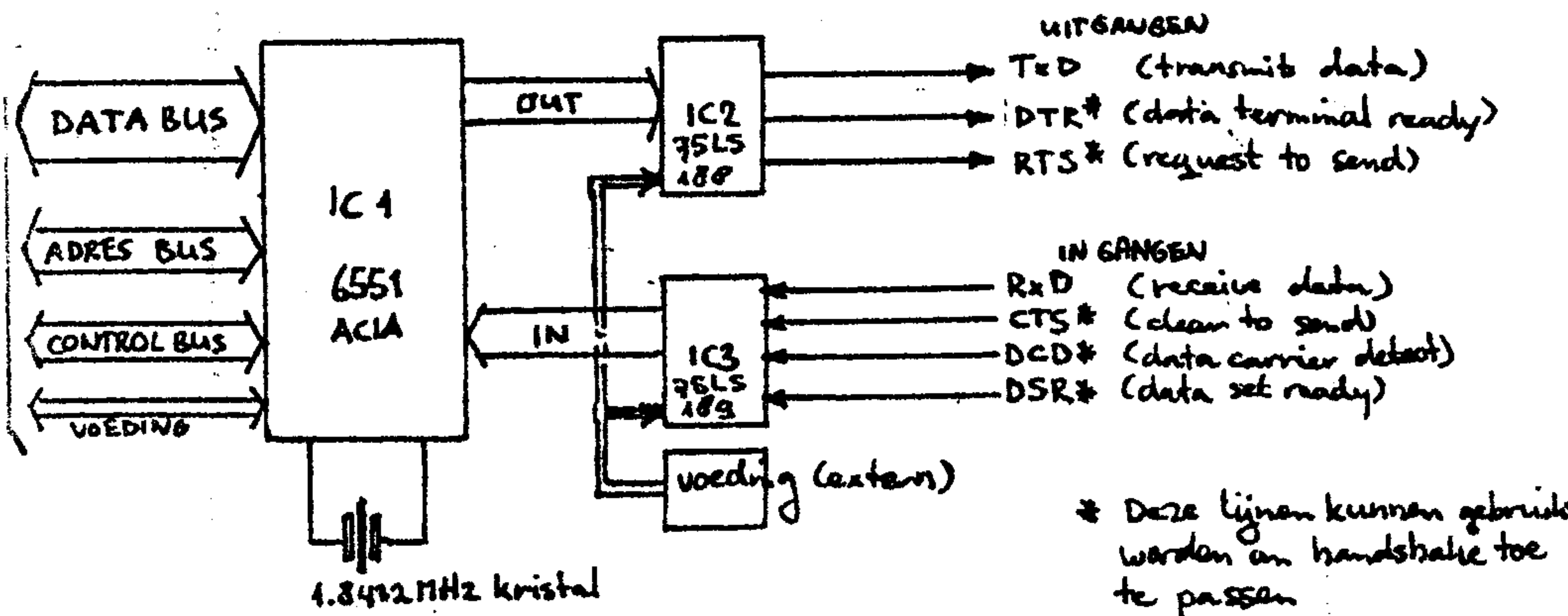
Ikzelf heb de schakeling nu in gebruik om met behulp van een MODEM per telefoon te communiceren met een andere computer die 40 km verder staat. Ook heb ik al een andere computer DIRECT gekoppeld om op de respectabele snelheid van 9600 Baud gegevens met elkaar uit te wisselen. En dan hebben we het nog niet eens over alle mogelijke randapparatuur.



Albert Verheij.

RS 232-C interface voor de ATOM.

Het verzenden van data tussen een computer en een randapparaat of andere computer kan op verschillende manieren. De eerste manier is parallel overdracht. In de Atom wordt daarin voorzien door het i.c. 6522 (V.I.A.) De tweede manier is serie overdracht. Deze methode wordt toegepast als er minder lijnen beschikbaar zijn of als er grote afstanden overbrugd moeten worden. Toepassing van een serie interface vindt men bij allerlei apparatuur zoals printers,beeldschermen,terminals,andere computers en MODEMS (voor data verzenden per telefoon).
Ook Viditel werkt met seriële overdracht.
De volgende schakeling is hiervoor ontworpen en kan op de interne connector PL8 van de ATOM aangesloten worden.



In het schema ziet U 1 grote i.c.,dat is de 6551 ACIA(Asynchronous Communications Interface Adapter). Dit i.c. bevat alle elektronica om data op de juiste manier tot te bewerken voor serie overdracht.
Deze seriële evenknie van de V.I.A. heeft net als de V.I.A. een aantal interne registers die geprogrammeerd kunnen worden.
Dat zijn achtereenvolgens de volgende registers:

register 0- Transmit/receive data register. Dit register bevat de ontvangen data als men het leest, en het bevat de te verzenden data als er in geschreven wordt.

register 1- Status register. Dit register bevat "vlaggen" ten behoeve van de programmeur.

register 2- Command Register. Dit register moet geprogrammeerd worden om de i.c. op de gewenste manier te schakelen.

register 3- Control register. Dit register moet men programmeren om de overdrachtsnelheid (baudrate) te bepalen.

Schema:

XTAL 1.8432 MHz

10 MΩ

Vcc

1kΩ

ongebruikte inverters

D7 D6 D5 D4 D3 D2 D1 D0

XTAL1 XTAL2

TxD DTR RTS RxD CTS DCD DSR

11x10kΩ

IC1 6551

CS1 CS0 RS0 RS1

A0 A1 A2 A3

N.C. A2

N.C. A3

RES IRQ Ø2 R/W

4x10K

Vcc GND

100nF

2x 68µF

LK1 LK2 LK3 LK4 LK5 LK6 LK7

330 pF

4x3,9kΩ

Transmit Data

Data Terminal Ready

Request To Send

Receive Data

Clear To Send

Data Carrier Detect

Data Set Ready

▬ = edge connector

1 = 74LS04

2 = 74LS04

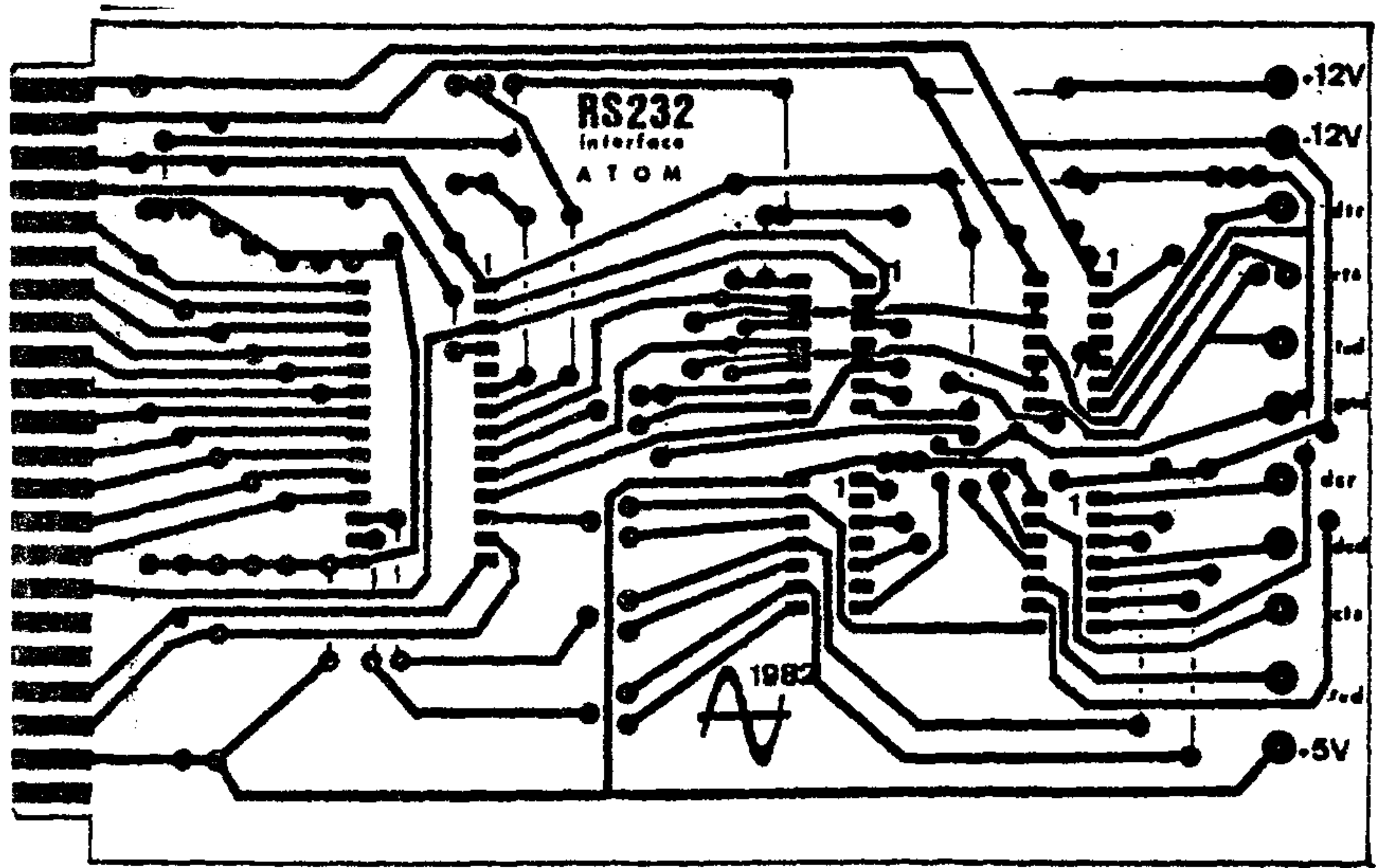
3 = 75188 /LM1488

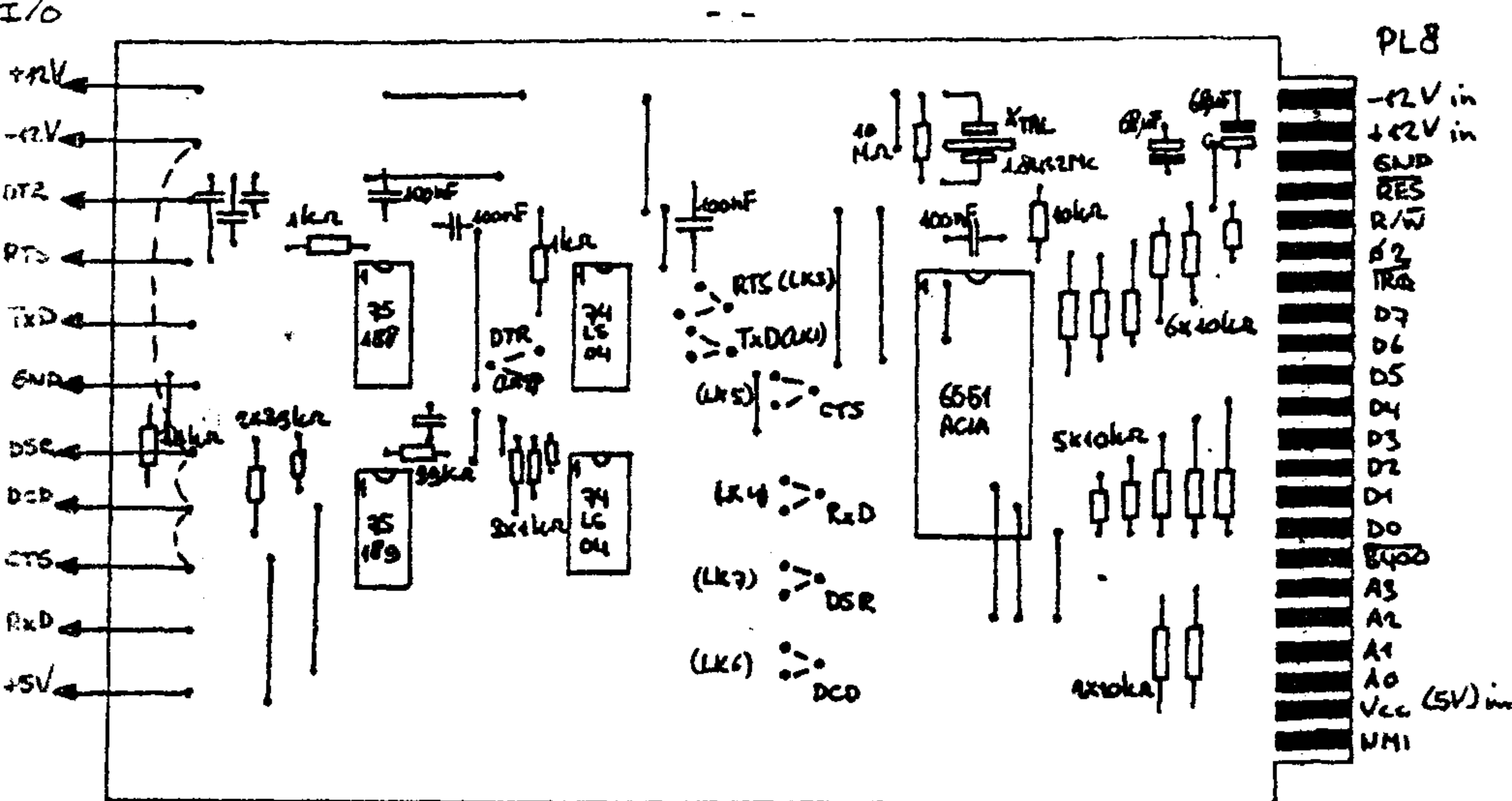
4 = 75189 /LM1489

LK1 t/m LK7 = keuze links om polariteit te bepalen

* inverteerd

PRINT LAY-OUT





Komponentenzijde

keuze verbinding (CTS ed.) doorverbinding met blank draad.

De edgeconnector is gebruikt omdat deze print als insteek-unit in een 19-inch rack wordt gebruikt. De connector kan worden verandert in een ander type dat voor inbouw in de ATOM geschikt is.

Let er overigens op dat er signalen met een potentiaal verbonden dienen te worden om goed functioneren van de ACIA te garanderen.
In het geval van mijn toepassing moet Clear to Send op de ACIA laag zijn omdat er anders niet gezonden kan worden. De beide signalen Data Set Ready en Data Carrier Detect moeten een potentiaal op de input hebben omdat er geen open ingang toegestaan is voor deze signalen.

De plaatsen waar de keuze-stropjes gemaakt dienen te worden ziet U in het midden van de print.

Let vooral goed op de aansluitingen van de print op de ATOM,kleine vergissingen kunnen grote konsekwenties hebben.

+++

Opmerkingen betreffende het statusregister: (zie pag.27)

De bits in het statusregister van de acia kunnen na in een bepaalde toestand gekomen te zijn gereset worden op de volgende manieren:

Parity error - wordt vanzelf gereset

Framing error - wordt vanzelf gereset

Overrun error - wordt vanzelf gereset

Rec.data reg. full - reset door uitlezen van rec.data reg.

Transm.data reg. empty - reset door schrijven in TxD reg.

$\overline{DCD}$ - niet resettable

$\overline{DSR}$ - niet resettable

$\overline{IRQ}$ - reset door uitlezen van status register

Programmatuur:

De ACIA is net als de VIA te programmeren door enkele bytes weg te schrijven in zijn interne registers. Bij de ACIA zijn dit slechts twee registers,een derde register bevat een STATUS WORD en een vierde register bevat de ontvangen of te verzenden data. De opbouw is als volgt:

- $B400 - Transmit data /Receive data Register
- $B401 - Status Register /Soft Reset
- $B402 - Command Register
- $B403 - Control Register

TROL REGISTER: 7 6 5 4 3 2 1 0

Bit 7:
0=1 stopbit
1=2 stopbits
1 stopbit (8-bits en pariteihit)
1½ stopbit (5-bits geen pariteit)

Bits 6-5:
0 0=8 bits
0 1=7 bits
1 0=6 bits
1 1=5 bits

Bit 4:
0=externe ontvanger klokpuls
1=interne baudrate-generator

Bits 3-0:

0	0	0	0	=16x externe klokpuls
0	0	0	1	= 50 BAUD
0	0	1	0	= 75 BAUD
0	0	1	1	= 110 BAUD
0	1	0	0	= 135 BAUD
0	1	0	1	= 150 BAUD
0	1	1	0	= 300 BAUD
0	1	1	1	= 600 BAUD
1	0	0	0	= 1200 BAUD
1	0	0	1	= 1800 BAUD
1	0	1	0	= 2400 BAUD
1	0	1	1	= 3600 BAUD
1	1	0	0	= 4800 BAUD
1	1	0	1	= 7200 BAUD
1	1	1	0	= 9600 BAUD
1	1	1	1	=19200 BAUD

MAND REGISTER: 7 6 5 4 3 2 1 0

Bits 7-5:
- - 0= geen pariteit bit
0 0 1= oneven pariteit
0 1 1= even pariteit
1 0 1= "1"pariteit verzonden geen controle.
1 1 1= "0"pariteit verzonden geen controle.

Bit 4:
0=normaal
1=echo(bits 2 en 3 moeten "0" zijn

Bits 3-2:

0 0=	RTS	hoog	transm.uit	TxD int.uit
0 1=	RTS	laag	transm.aan	TxD int.aan
1 0=	RTS	laag	transm.aan	TxD int.uit
1 1=	RTS	laag	transm.BRK	TxD int.uit

Bit 1:
0= IRQ interrupt aan via bit 3 ST.R
1= IRQ interrupt uit

Bit 0:
0=ontvanger en alle interrupts uit (DTR hoog)
1=ontvanger en alle interrupts aan (DTR laag)

RESET:

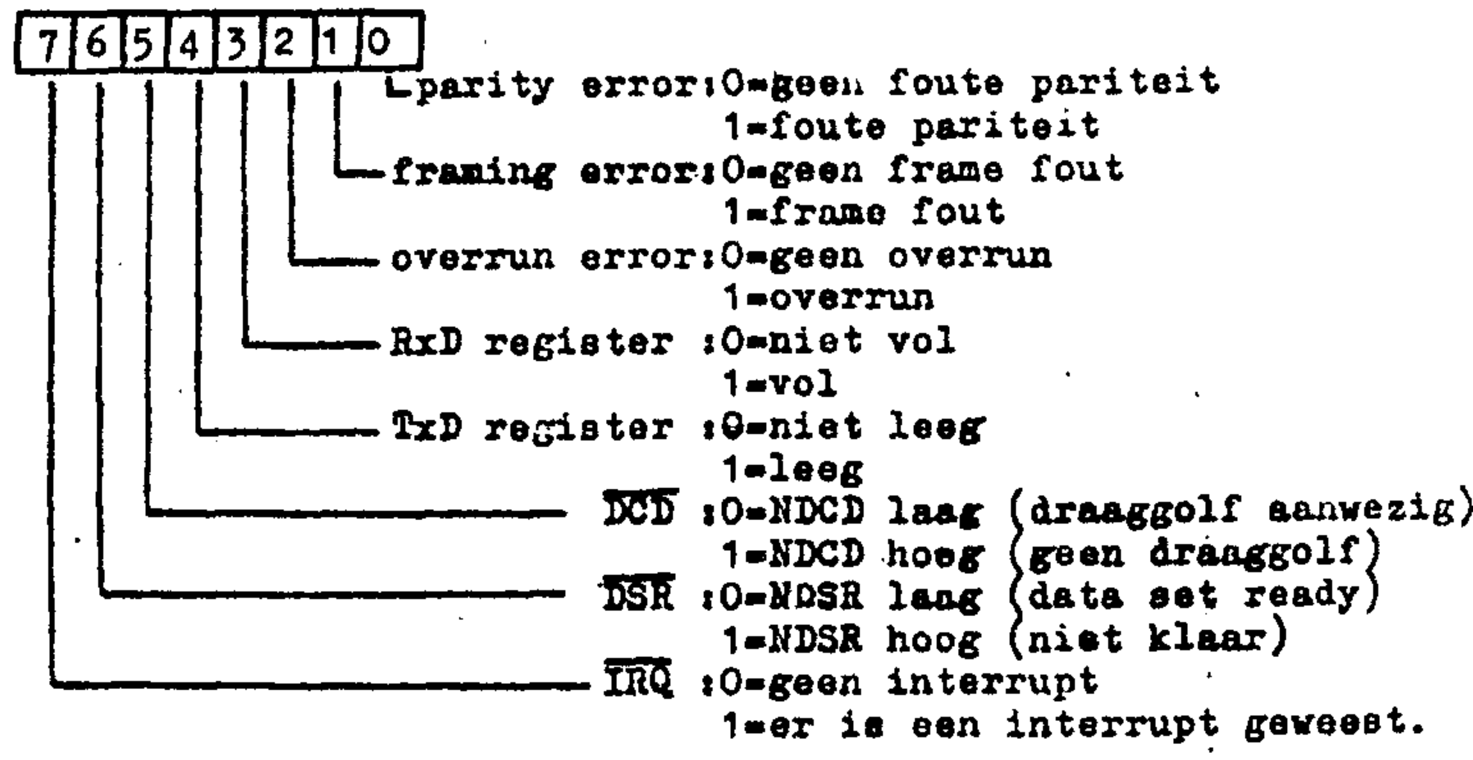
(CONTROL REGISTER)

	7	6	5	4	3	2	1	0
hardware reset	0	0	0	0	0	0	0	0
software reset	-	-	-	-	-	-	-	-

(COMMAND REGISTER)

	7	6	5	4	3	2	1	0
hardware reset	0	0	0	0	0	0	0	0
software reset	-	-	-	0	0	0	0	0

STATUS REGISTER:



bij reset:

7	6	5	4	3	2	1	0	
0	-	-	1	0	0	0	0	hardware reset
-	-	-	-	-	0	-	-	software reset

Toepassingen:

In mijn toepassing is de RS 232-C interface ingebruik om met behulp van een MODEM (modulator-demodulator) data via telefoonlijn over te zenden. In dit geval heb ik het NCTS-signaal aan -12V gehangen om te kunnen zenden. Nodig is dan natuurlijk de modem. bv.een 300BAUD asynchrone modem met P.T.T. goedkeuring. Andere toepassingen zijn bv. het besturen van printers, plotters, video-terminals, papier-terminals, andere typen computers enz. Over het algemeen worden dit soort apparatuur vaak uitgerust met RS 232-C interfaces. Persoonlijk heb ik mijn ATOM al enkele malen direct of via modem verbonden met een andere computer. In dit geval was de andere computer een D.A.I. micro met de 8080 als processor. Deze verbinding werkte uitstekend en de mogelijkheden zijn erg groot. Twee computers kunnen informatie van hun toetsenborden zichtbaar maken op hun eigen scherm en op het scherm van de partner. Ook kan informatie uit het geheugen worden gecopieerd en aangepast.
Twee ATOM's onderling kunnen op hoge snelheid (max.19200 baud) programma's uitwisselen. Een verdere mogelijkheid is het aansluiten op een telex (50 baud,5-bits code) zonder noemenswaardige problemen.

Noodzaak blijft natuurlijk altijd om een stuur(driver)programma te schrijven waarmee de interface werkt.Een dergelijk programma kan in BASIC,of als dit te traag is in machinetaal geschreven worden.
Bijgevoegd vindt U enkele nuttige routine's en een programma om een modem te bedienen.

Stuurprogramma: (voorbeeld)

```
 10 P.$12
 20 P."RS-232-C DRIVER"'
 30 P."8-1982 A.Verheij"'
 40 P."---------------------------"'
 50 DIM LL(4),P(-1)
 60 GOS.a
 70 P."DE ATOM IS NU EEN DATATERMINAL"'
 80 P."CONTROL-CODES: CTRL-H BACKSPACE"'
 90 P."               CTRL-I HOR.TAB"'
100 P."               CTRL-J LINEFEED"'
110 P."               CTRL-K VERT.TAB"'
120 P."               CTRL-L FORMFEED"'
130 P."               CTRL-M CARR.RET."'
140 P."               CTRL-[ ESCAPE"'
150 P."               CTRL-↑ HOME CURS"'
160 LINK LL0;REM INITIALISEER ACIA
170 DO
180 LINK LL1;REM SCAN RS-232 EN KEYBOARD
190 UNTIL 0
200 END
210aP.$21
220[
230:LL0 LDA@#16  /control word in accu
240 STA#B403     /accu naar control register
250 LDA@#0A      /command word in accu
260 STA#B402     /accu naar command register
270 RTS          /return
280:LL1 LDA@#08  /masker in accu
290 BIT#B401     /test bit 3 van het status register
300 BNE LL2      /indien ongelijk naar LL2
310 LDA#B400     /inhoud receive register in accu
320 JSR#FFF4     /druk inhoud van accu af
330:LL2 JSR#FE71 /is er een toets ingedrukt
340 BCS LL3      /indien carry (geen toets) ,return
350 JSR#FFE3     /haal toets op en zet in accu
360 JSR#FFF4     /druk inhoud accu af
370 STA#B400     /breng inhoud accu naar transmit register
380:LL3 RTS      /return
390];P.$6;RETURN
```

Specificaties:

voeding:	+12 V/-12 V/+5 V/0 V(GND)
inputs/outputs:	+12 V/-12 V t.o.v.GND
baudrates:	50,75,110,135,150,300,600,1200,1800,2400,3600,4800,7200,9600,19200 Bd.
externe klok:	16 x baudrate (optie op pen RxC van 6551)
kristal:	1,8432 MHz
stropjes:	alle inputs/outputs hebben polariteit-keuze
handshake:	aanwezig zijn: CTS- Clear To Send (input) RTS- Request To Send (output) DSR- Data Set Ready (input) DTR- Data Terminal Ready (output) DCD- Data Carrier Detect (input)

Prijsindicatie: (onderdelen bij de gewone vakhandel gekocht)

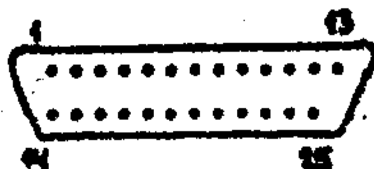
6551 ACIA	ƒ 50,-
1488 Line Driver	ƒ 4,-
1489 Line Rec.	ƒ 4,-
74LS04 inverter	ƒ 1,- (2x)
Kristal	ƒ 25,-
D-connector	ƒ 30,- (incl.kapje)
Flatcable	ƒ 5,-

Totaal is de interface te bouwen voor ongeveer ƒ120,- (als de print ong. ƒ15,- kost)
De print is enkelzijdig te maken als men geen bezwaar hoeft tegen een aantal stropjes.
De voeding voor de interface zal inclusief trafo ongeveer ƒ35,- kosten.

Pin Configuratie ACIA en D-connector:

SY6551 ACIA

1	GND	R/$\overline{W}$
	CS0	ϕ2
	$\overline{CS1}$	$\overline{IRQ}$
	$\overline{RES}$	DB7
	RxC	DB6
	Xtal1	DB5
	Xtal2	DB4
	$\overline{RTS}$	DB3
	$\overline{CTS}$	DB2
	TxD	DB1
	$\overline{DTR}$	DB0
	RxD	$\overline{DSR}$
	RS0	$\overline{DCD}$
15	RS1	Vcc



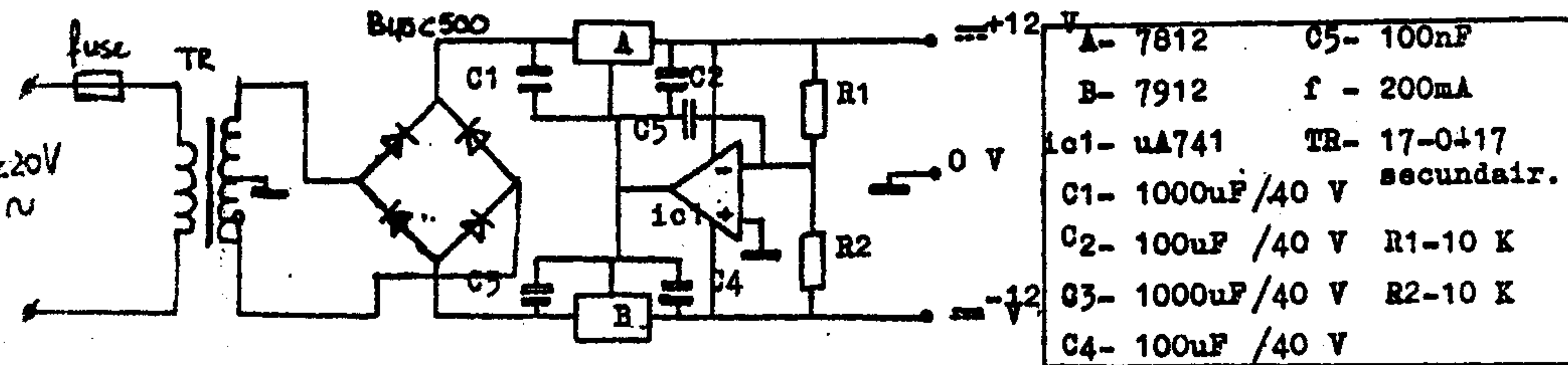
TxD - pen 2
RxD - pen 3
GND - pen 7
CTS - pen 5
RTS - pen 4
DTR - pen 20
DSR - pen 6

De andere pennen zijn voor zover mij bekend is niet benoemd, dus kan men de andere signalen daarop aanbrengen.

De Voeding:

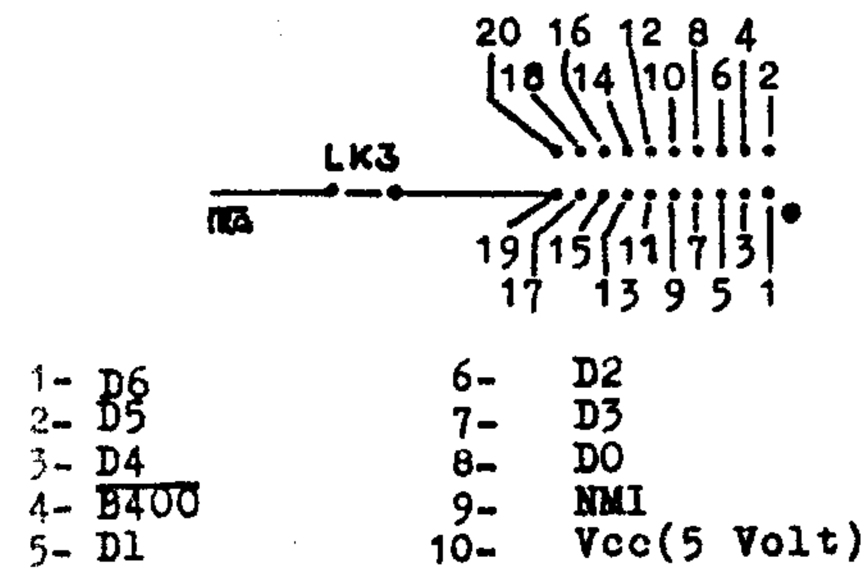
De voeding die toegepast wordt kan eenvoudig gehouden worden. Er worden geen hoge eisen gesteld aan de symmetrie van deze voeding.
Een goed werkende voeding vindt u op het onderstaande schema, waarbij toch een mooie symmetrische voedingsspanning voor de interface en evt.andere schakelingen wordt verkregen. De toegepaste OPAMP houdt de uitgangsspanning mooi stabiel met zeer kleine verschillen tussen de positieve en de negatieve voeding.

Schema (voeding):



Opmerkingen:
Zoals U waarschijnlijk al gezien hebt wordt bij het aansluiten op de ATOM gebruik gemaakt van de vrije plaats voor een connector PL8. Doordat de print van het Eurokaart formaat is past deze gemakkelijk in de ATOM-kast. Voor de 25-pol. D-connector zal een gat in de achterwand van de kast gemaakt moeten worden.
(bv. onder de Extension Bus).

De pinconfiguratie van PL8 staat beschreven in de bouwbeschrijving van de ATOM
n is als volgt:



(gezien vanaf kant 1 van de print)

LK 3 moet aangebracht worden teneinde IRQ mogelijk te maken.

1-	D6	6-	D2	11-	GND	16-	A2
2-	D5	7-	D3	12-	A1	17-	Ø2 clock
3-	D4	8-	D0	13-	D7	18-	R/W̄
4-	B̄4̄0̄0̄	9-	NMI	14-	A0	19-	ĪR̄Q̄
5-	D1	10-	Vcc(5 Volt)	15-	A3	20-	R̄S̄T̄

Ab Verheij

Geachte heer Borghaerts:

Hierbij een programma'tje voor het tekenen met de Joy-stick.Het programma heeft de volgende voordelen t.o.v. dat van u (acorn nieuws 5) :

1) Als de Stick heen en weer bewogen wordt, wordt er niet direct een lijn getrokken, maar gaat er slechts een punt heen en weer; tekenen geschiedt na het indrukken van de knop.

2) Tekeningen kunnen worden opgeslagen op band; na teruglezen werkt programma nog, en de tekening kan dan eventueel nog verbeterd worden.

HET WERKEN MET DIT PROGRAMMA:

De tekening, die we uiteindelijk op het scherm willen hebben,tekenen we met een (erg) zacht potlood of vilstift op een stuk doorzichtig plastik, wat we daarna op het scherm bevestigen.(we kunnen natuurlijk ook direct op onze T.V. tekenen, maar dit heeft weer als nadeel dat we het scherm telkens schoon moeten maken, hetgeen (vreemd genoeg) niet met P.$ 12 lukt.)

We bewegen de stip nu naar een lijn, die we na het indrukken van de knop over kunnen trekken.We zullen merken dat we geen schuine lijnen kunnen trekken, wat we oplossen door: Joy-stick omhoog,rechts,omhoog...etc, dat (hoe vreemd dit ook klinkt,)in de praktijk beter blijkt te werken.

Als we dan uiteindelijk tevreden zijn over ons kunstwerk kunnen we hem opslaan: na het intypen van een S (ja,ja, inderdaad van Save) kunnen we de recorder starten (RECORD TAPE), en de spatiebalk indrukken.
De tekening wordt dan geruisloos, zonder dat wij er ook maar iets van merken,ruiken of zien, overgeseind.

Terugroepen van een tekening geschiedt na het indrukken van een L(oad), starten van band (PLAY TAPE), en indrukken van de spatiebalk.
(opmerking:deze beide handelingen worden verricht zonder naam; wat het voordeel heeft veel sneller te zijn, maar het nadeel heeft dat een tekening meer makkelijk kan "verdwalen" ergens op een band,oplossing:tellerstand van recorder goed bijhouden.;anders regel 210&240 veranderen.)
Het programma spreekt eigenlijk verder voor zich, ieder kan aanpassingen maken naar zijn behoefte (wit op zwart ipv. andersom, stippelen, voorgeprogrammeerde cirkels ,vierkanten e.d., etc. etc.)



PROGRAMMA JOYSTICK

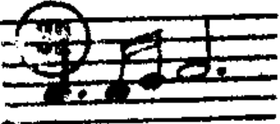
Andre' de Bruin

```
10   DIM P(-1);B=#B001;X=0;Y=0
20    [;JSR#FE71; STY#80; RTS;]
30   CLEAR 4
40   FOR Q =#8000 TO#9800  STEP 4
50   !Q=-1; NEXT Q
60   *NOMON
70   LINK TOP; R=?#80;IF R= 51 GOSUB 200
80   IF R= 44 GOSUB 230
90   IF ?B%2 =0 PLOT15,Z,Y ; GOTO 110
100  PLOT 13,X,Y
110  IF ?B = 255 PDOT 15,X,Y; A=1 ;GOTO 70
120  IF A = 1 PLOT 13,X,Y ; A = 0
130  IF ?B = 239 OR ?B = 238 Y=Y+1
140  IF ?B = 247 OR ?B = 246 X=X+1
150  IF ?B = 251 OR ?B = 250 Y=Y-1
160  IF ?B = 253 OR ?B = 252 X=X-1
170  FOR I = 1 TO 1 ; NEXT
175  PLOT 15,X,Y; GOTO 70
200  P. $ 30
210  *SAVE 8000 9800
220  RETURN

230  P. $ 30
240  *FLOAD 8000
250  RETURN
```

REM: regel 170 regelt snelheid van lijntrekken; I=1 to 100 is erg praktisch.
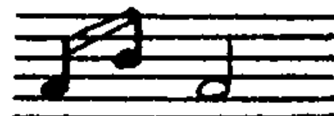




muziek uit het noorden

Wim Schreuder uit Haren kwam op het idee n.a.v. 'harpsichord' een programma te schrijven, dat het mogelijk maakt méér tonen te programmeren en de toonlengte per toon te regelen.
Voor iedereen, die een TOOLKIT heeft, die de statements BEEPX,Y,READ en DATA bevat, zijn hier twee programma's .

++++ M U Z I E K D O O S+++++



```
 10  =0
 20  P.$12
 30  DIMAA(475),BB(475)
 40  P."HET AANTAL TONEN WORDT STEEDS"' "BIJGEHOUDEN,HET MAXIMUM"'
 50  P."IS 475"'"ALS U UW LAATSTE TOON INGEVOERD HEBT,"'
 65  P."TYPE DAN VOOR HOOGTE EN LENGTE EEN'0' IN"'
 70  P."DE TUNE WORDT DAN GESPEELD"'"VEEL PLEZIER !!"'
 80  P."DRUK OP EEN TOETS"'
 90  LINK  FFE3;P.$12
100  A=-1
110  DO
150   A=A+1
160   P."("A")"'
170   IN."TOONHOOGTE"B
180   IF B=0 THEN C=A
190   IN "TOONLENGTE"D
200   AA(A)=B;BB(A)=D
210  UNTIL B=0
220  A=-1
230  DO
240   A=A+1
250   B=AA(A); D=BB(A)
260   BEEPB,D
270  UNTIL A=C;P.$12
280  P."TUNE NOG EENS SPELEN?"'
290  IN."(JA=1,NEE=0)"Z
300  IF Z=1 GOTO 220
310  P.$12
320  P."HOPENLIJK VOND U HET LEUK"'
330  END
```



```
 10  X=-1;P.$12;RESTORE
 20  DO
 30    X=X+1
 40    P."::::WILHELMUS VAN NASSOUWE::::"'
 50  UNTIL X=16
 60  READX;READ Y
 70  IF X=1111 GOTO 500
 80  BEEPX,Y
 90  DATA 183,13,137,13,137,13,121,6,108,6,102,
          6,121,6
100  DATA 108,13,121,6,108,6,102,13,108,13,121,
          6,137,6
110  DATA 121,13,137,37,183,13,137,13,121,6,108,6
120  DATA 102,6,121,108,13,121,6,108,6,102,13,
           108,13
130  DATA 121,6,137,6,121,13,137,37,108,6,102,6,
          91,25
140  DATA 81,13,91,25,102,13,108,13,121,6,108,6,
          102,13
150  DATA 108,13,121,13,137,13,121,25,183,13,
          6,145,6
170  DATA 163,6,145,6,137,13,137,13,145.13,137,37
180  DATA 1111,0
 190 GOTO 60
500  IN."TUNE NOG EENS SPELEN?(JA=1)"A
510  IF A=1 GOTO 10; END
```

TOONHOOGTE	oktaaf 1	oktaaf 2	oktaaf 3	oktaaf 4
DO C		205	102	50
C		194	96	47
RE D		183	91	45
D		173	86	42
MI E		163	81	39
FA F		154	76	37
F		145	72	35
SO G		137	68	33
G		129	64	31
LA A	246	121	60	29
Bb	231	114	57	
TI B	217	108	54	26
DO C				24

T O O N L E N G T E

Sec.	GETAL
5	250
2	100
1	50
½	25
1/4	13
1/8	6
1/16	3

Spatie: van b.v. 2 sec:
BEEP X,Y, waarbij:
Toonhoogte = 1
Toonlengte = 100

W.A. Schreuder.

PRIEMGETALLEN

Berekent alle priemgetallen groter dan een opgegeven beginwaarde.

```
L.
 5  ClearO
 10 Print "Priemgetallen"' '
 20 Input "beginwaarde: "K;print'
 30 If K%2=0 then K=K-1
 40 K=K+2;D=1
 50 D=D+2;H=K/D;L=K%D
 60 If D>H Then print K," is een priem-
    getal."';goto 40
 70 If L=0 goto 40
 80 Goto 50
 90 END

 RUN
 Priemgetallen
 Beginwaarde: ?250
      251 is een priemgetal
      257 is een priemgetal
      263 is een priemgetal
      269 is een priemgetal
      271 is een priemgetal
      277 is een priemgetal
```

++++++++++++++++++++++++++++++++++++++

LOTTO-HULPJE

Het programma doet wat het zegt: Het verzint getallen, die je kan invullen op je lotto-formulier en zet ze vervolgens op volgorde van laag naar hoog.

```
L.
 10 DIM H(5)
 20 DIM ZZ(6)
 22 Input "Geef een getal"X
 24 For N=1 to X
 26 Y=ABS(RND%41)+1
 28 Next N
 30 ZZ(1)=ABS(RND%41)+1
 50 For N=2 to 6
 60 ZZ(N)=ABS(RND%41)+1
 70 For M=1 to N-1
 80 If ZZ(N)=ZZ(M) goto 60
 90 Next M
110 Next N
120 For N=1 to 5
130 For M=N+1 to 6
140 If ZZ(M) ZZ(N) goto 200
150 Y=ZZ(N)
160 ZZ(N)=ZZ(M)
170 ZZ(M)=Y
200 Next M
210 P.ZZ(N)
220 Next N
225 P.ZZ(6)
230 END

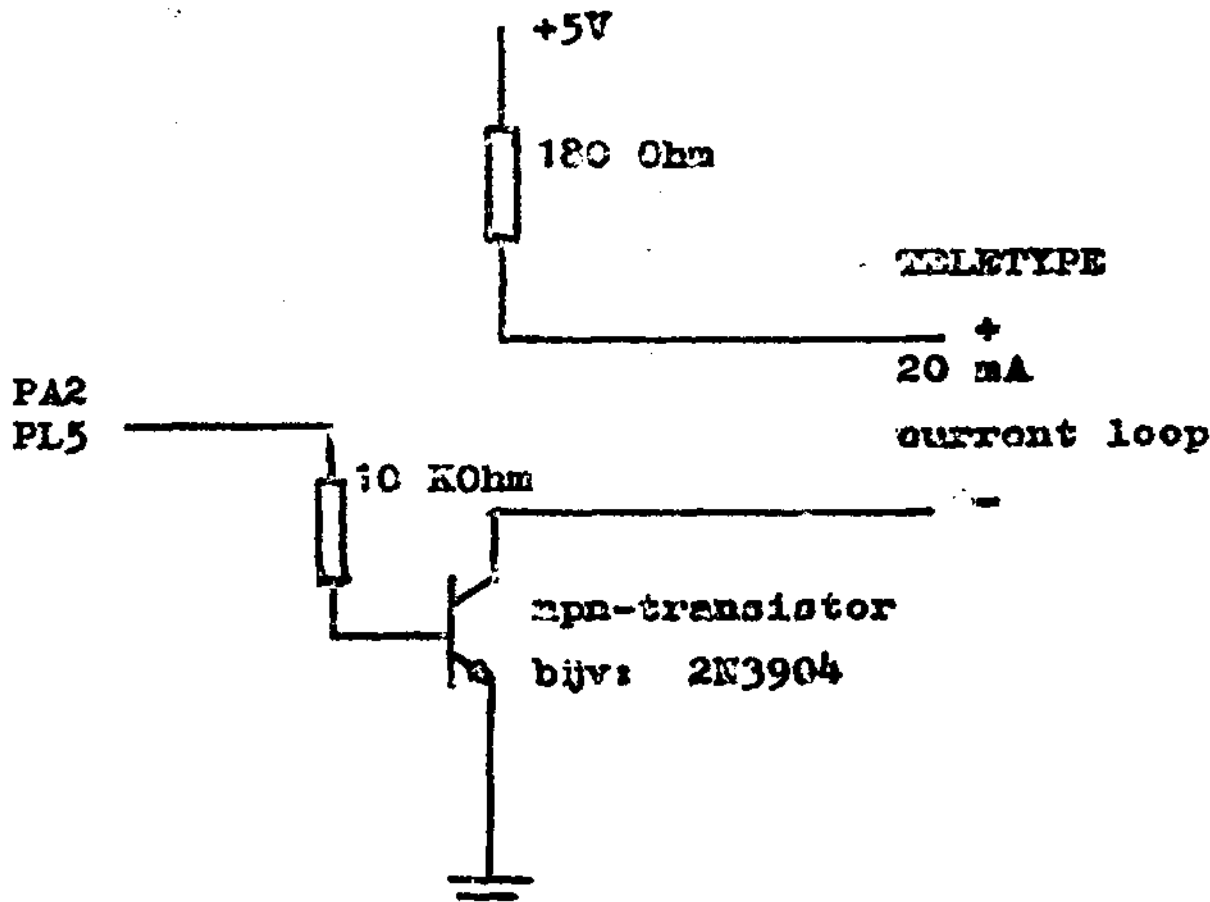
RUN
Geef een getal?23
 4
10
14
22
36
39
```

++++++++++++++++++++++++++++++++

T E L E T Y P E - I N T E R F A C E

In Acorn Nieuws nr. 3 blz 21 was al eens zo'n interface gepubliceerd. De nand-poort van de 7438 hoeft echter alleen te schakelen tussen wel en geen stroom. DAT KAN EEN ENKELE TRANSISTOR OOK !! Vandaar dat ik zelf de scakeling gebruik, die hierbij is afgedrukt. Deze heeft de volgende voordelen: 1) Hij is goedkoper 2) Hij is kleiner, dus gemakkelijker in te bouwen (bij mij zit hij gewoon op de printer-connector gesoldeerd.).
Overigens: vrijwel ieder type npn.transistor zal in deze schakeling voldoen.

G. Akkermans.



n i m s p e l


```
L.1,570
 10 @=10:Z=1:REM VOOR TOOLBOX:Z=1, ANDERS Z=0
 20 P.$12
 30 P."NIMSPEL"''
 40 ?#E1=0
 50 DIM BB(3)
 60 DIM VV(4)
 70 DIM H(4)
 80 DIM Q(20)
 90 VV(1)=8:VV(2)=4:VV(3)=2:VV(4)=1:
100 INPUT"KEN JE DE SPELREGELS (JA/NEE)"$H
110 IF ?H=78 GOSUB 780
120 INPUT"HOE SPEEL JE?(0=GOED,1=MATIG,   2=SLECHT)"S
130 P.'
140 INPUT"WAT IS JE NAAM"$Q
150 FOR N=1 TO 3:BB(N)=ABS(RND%15)+1:NEXT N
160 P.'
170 INPUT"WIL JE BEGINNEN? (JA/NEE)"$H
180 T=1:Y=1:GOSUB 1090
190 IF ?H=74 GOSUB 580
200 IF T=0 GOTO 320
210 IF S=0 GOTO 230
220 IF E=0 THEN P. "DAT WAS EEN GOEDE ZET,"',$Q,"!"'
230 IF Y=1 GOTO 250
240 IF E<>0 THENP. "DAT WAS EEN SLECHTE ZET,"',$Q,"!"'
250 F.N=1TO20 :WAIT:N.N
270 GOSUB 490
280 IF T=0 GOTO 320
290 Y=0
300 IF E=0 THEN Y=1
310 $H="JA":GOTO 190
320 INPUT"NOG EEN SPELLETJE? (JA/NEE)"$H
330 IF ?H=74 GOTO 150
340 P.$12,"OM TE SPELEN: TYP 'RUN'!"'
350 END
360 GOSUB 1110
370 IF E=0 GOTO 770
380 IF ABS(RND%5)<S GOTO 740
390 N=0
400 N=N+1
410 IF (E&VV(N))=VV(N) GOTO 430
420 GOTO 400
430 I=0
440 I=I+1
450 IF (BB(I)&VV(N))=VV(N) GOTO 470
460 GOTO 440
470 BB(I)=BB(I):E .
480 RETURN
490 P.    "MIJN BEURT"
500 F.N=1TO100:WAIT:N.N
510 GOSUB 360
520 FOR N=1 TO 10  :WAIT:NEXT N
530 GOSUB 1080
540 IF T=0 P."IK HEB GEWONNEN!"''"JAMMER VOOR JOU, ",$Q''
550 IF Z=0 GOTO 570
560 IF T=0 F.I=1TO5:F.N=80TO40STEP-4:BEEPN,1:N.N:N.I
570 RETURN
```

```
L.580,1130
 580 P."JOUW BEURT"'
 590 INPUT"WELKE STAPEL"I
 600 IF I<1 GOTO 590
 610 IF I>3 GOTO 590
 620 INPUT"HOEVEEL ERAF"L
 630 IF L<1 GOTO 620
 640 IF Z=0 GOTO 660
 650 IF BB(I)<L P."DIT KAN NIET!"'':BEEP 44,44:GOTO 590
 660 IF BB(I)<L P."DIT KAN NIET!!!"'':GOTO 620
 670 BB(I)= BB(I)-L
 680 GOSUB 1080
 690 IF T=0 P.''"JE HEBT GEWONNEN!"',"GEFELICITEERD, ",$Q''
 700 IF Z=0 GOTO 720
 710 IF T=0 F.I=1TO7:F.N=40TO80STEP4:BEEPN,1:N.N:N.I
 720 RETURN
 740 N=ABS(RND%3)+1
 750 IF BB(N)>S BB(N)=BB(N)-S-1:GOTO 480
 760 IF BB(N)>0 BB(N)=BB(N)-1:GOTO 480
 770 GOTO 740
 780 P.$12
 790 ?#E1=0
 800 P.   "HET NIMSPEL IS EEN GEZELSCHAPS-"'
 810 P."SPEL, DAT IN ZIJN EENVOUDIGSTE"'
 820 P."VORM GESPEELD WORDT DOOR "'
 830 P."TWEE SPELERS."'
 840 P."EEN IN PRINCIPE WILLEKEURIG"'
 850 P."AANTAL FICHES IS VERDEELD OVER"'
 860 P."DRIE STAPELTJES."'
 870 P."OM DE BEURT NEEMT NU EEN SPELER"'
 880 P."VAN EEN DER STAPELTJES EEN"'
 890 P."AANTAL FICHES WEG! DEGENE DIE,"'
 900 P."OP DEZE WIJZE VOORTSPELENDE,"'
 910 P."DE LAATSTE FICHE WEGNEEMT IS"'
 920 P."WINNAAR!"''
 930 INPUT"DRUK OP RETURN"$H
 940 P.$12
 950 ?#E1=0
 960 P."IN DEZE UITVOERING NEEMT DE"'
 970 P."COMPUTER DE ROL VAN EEN VAN DE"'
 980 P."SPELERS OVER."'
 990 P."DE COMPUTER VRAAGT ZELF AAN U"'
1000 P."VAN WELKE STAPEL EN HOEVEEL"'
1010 P."FICHES U WILT WEGNEMEN!"'
1020 P."DE COMPUTER GEEFT DE AANTALLEN"'
1030 P."FICHES WEER DOOR DEZE NAAST"'
1040 P."ELKAAR AF TE DRUKKEN."'
1050 P.''
1060 P."VEEL SUCCES!"''
1070 RETURN
1080 P.'"DIT IS HET RESULTAAT:"'
1090 P.'"STAPEL1   STAPEL2 ' STAPEL3"
1100 P.BB(1),BB(2),BB(3)''
1110 E=BB(1):BB(2):E=E:BB(3)
1120 T=BB(1)+BB(2)+BB(3)
1130 RETURN
```

Het is een denkspelletje dat je kan spelen tegen de computer.

Gerard Akkermans

1 → 2 → 4 MHz ?

Door Arno Millenaar.

Dit artikel gaat over hoe men met eenvoudige middelen de ACORN soft/hardware matig kan omschakelen tussen een kloksnelheid van 1MHZ en 2MHZ.
In de eerste experimentele fase monteer je een wisselschakelaar,zodat je al even kunt merken wat het verschil is in verwerkings snelheid. Hierdoor wordt vanzelf warm voor beter werk wat hierna volgt, maar waar wel een paar uurtjes knutselen voor nodig is.

Het speelt zich voornamelijk af rond IC 44 de 74LS393. Dit IC deelt het 4 MHZ kloksignaal zodanig dat in de originele uitvoering de 6502 CPU een 1 MHZ signaal krijgt.

Wil je het met een wisselschakelaar doen (zie fig. 1) dan is dit wel leuk om even te proberen maar blijvend zal het geen succes zijn en wel om twee redenen:

a. Na elke omschakeling is er een "BREAK" nodig.Dit komt door het denderen en het niet synchroon met de klokpuls schakelen, toevals treffers uitgezonderd natuurlijk.

b. Het is niet mogelijk om vanuit het programma de snelheid te wijzigen. (soft schakelen)

De volgende eenvoudige schakeling (fig.II) voldoet al beter en heeft de twee voorgaande nadelen niet, De schakeling heeft slechts één IC nodig en wel de 7400.
Toch is er ook hier een nadeel en wel dit: na het aanzetten of na een BREAK weet je niet of je in de 1 MHZ- of 2 MHZ mode bent. Dit is te kontrolleren d.m.v. "CTRL G" (vergeet niet hierna op ESC te drukken).In dien de toon twee maal zo hoog is als normaal dan is het uiteraard de twee MHZ mode.
Deze schakeling maakt gebruik van een uitgangspoort van IC 25. Door de select-lijn aan de PC-3 poort van de 8255 PIA te hangen is het mogelijk om vanaf het toetsenbord of vanuit een programma d.m.v. de volgende instructies de snelheid te veranderen zonder deze door een BREAKte moeten resetten.

    ? ≠ B002= ≠ EF; voor de 2 MHZ mode
    ? ≠ B002= ≠ E7; voor de 1 MHZ mode   (≠ = "hex" teken)

Het gebruik van deze poort PC-3 behoeft nog wel nader kommentaar maar daarover later.
Een leuk demonstratie programma geeft heel duidelijk de verandering van snelheid aan.

```
DO P.$7;?≠B002 = ≠ EF;P.$7;?≠B002 = ≠ E7;U.0
```

Het resultaat zal een wisselende toonhoogte zijn.

Het navolgende schema (fig. III) kent geen problemen.Bij elke BREAK komt hij terug in de 1 MHZ mode.Indien hij terug komt in de 2 MHZ mode dan moet je de draadjes bij pin 11 en 13 van IC 44 verwisselen.
Hiervoor zijn twee IC's nodig, kosten ongeveer FL.3,50 en wel de 7486 (EXOR) en een 7473 (een JK flip flop).
Doordat er pas geschakeld wordt op de flank van de klok kan er geen fase fout worden gemaakt, ook ontstaan er geen reset problemen meer.Om deze schakeling te gebruiken heb je weer een poort nodig van 8255 PIA zoals dat ook bij de vorige schakeling gebeurde,
Om hierop terug te komen, ik ben mij bewust van het feit dat ik hier met mijn vingers in de standarisatie van de computer zit maar ik zag hier geen andere mogelijkheid of men moet een tweede 8255 monteren.Maar dit zou voor veel mensen een te grote stap zijn en hen ervan weerhouden omdit alles eens te proberen.
Als poort PC-3(pin 17 van IC 25) gebruikt wordt voor andere doeleinden dan die welke ACORN-computers hem heeft toebedeeld (het omschakelen van achterkleuren en contrast) dan moet LK 5 op de print (componentenzijde) worden doorgekrast.

Om een onrustig beeld te voorkomen moet je pin 39 van IC 31(6847) met een "pull-up" weerstand van ± 6k8 aan de + 5V hangen.
Waaneer je dit doet wordt het contrast van de computer ook sterker.Als dat niet gewenst is moet je pin 39 d.m.v LK 6 aan de massa leggen.

Ik heb dit alles al bij twee ATOM's gedaan en kwam hierbij het volgende probleem tegen. Het is niet gezegd dat dit altijd voorkomt want dat ligt aan de snelheid van de 2114 geheugen IC's van het video geheugen.In de mode 2 tm. 4 (CLEAR 2tm. 4) wil het bij plotten vaak een rommeltje geven.
Hiervoor zijn twee oplossingen:
a. De IC's vervangen door snellere types (heb ik zelf gedaan) en eventueel meten of de IC's wel hun broodnodige 5 volt krijgen.
b. Door voor elke PLOT,DRAW..... instructie de snelheid terug te brengen naar 1 MHZ en erna direkt weer omhoog naar 2 MHZ.
Een voorbeeld hiervan:

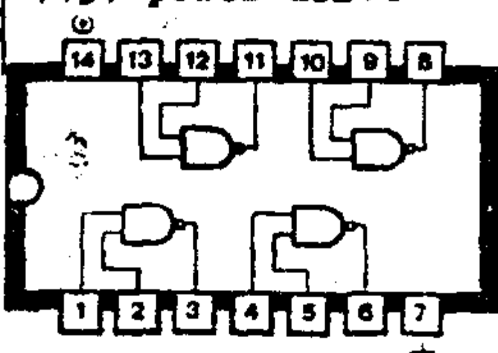
```
regelnummer   ?≠B002=≠E7;DRAW X,Y;?≠B002=≠EF
```

-of elke andere instructie-

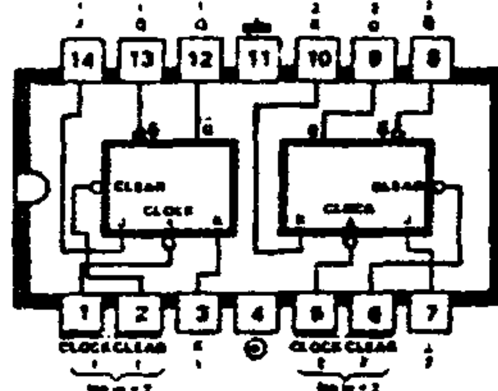
Bij dit alles is er wel een maar!.
De temperatuur loopt op en de stroom consumptie neemt toe.Dat zijn maar kleinigheden zolang je erop let dat de IC's niet boven de 100° komen(max. is 120° C.).
Indien je geen geforceerde koeling gebruikt moet je de computer gewoon niet onnodig lang in de 2MHZ mode laten staan.
Wat die koeling betreft, hier een suggestie:
Bij de KWANTUM HALLEN (zwolle, waarschijnlijk ook elders) wordt een axiaale kooi ventilator verkocht welke werkt op 12 volt, maar ook wel op 5 volt(eventueel de ongestabiliseerde spanning gebruiken).Deze ventilator kost maar FL. 6,50 en is op de zijkant van de kunststof kast van de ATOM te monteren.
Hiervoor moet je dan wel een langwerpig gat maken van ± 10 cm bij 2 cm en aan de aandere kant een rooster of een paar gaatjes.
Als je zoals ik de ATOM niet meer in de originele kast hebt dan kun je de bij de ventilator geleverde beugel gebruiken zodat montage nog eenvoudiger is.
Veel succes.

Quadruple 2 Input
NAND Gates

7400
7403 open collect. outp.
7437 power drive

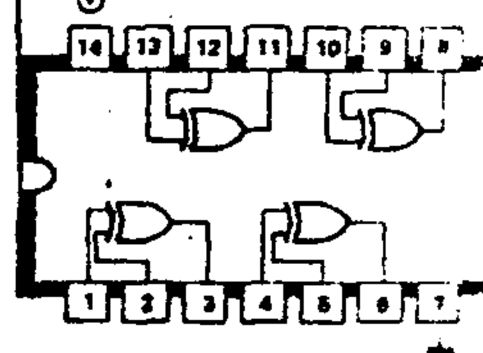


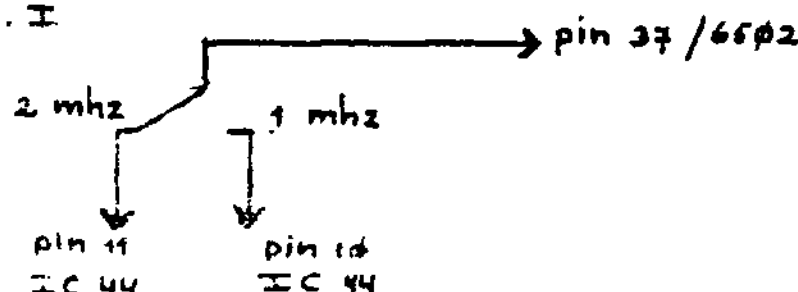
Dual JK FLIP FLOP
with clear
7473

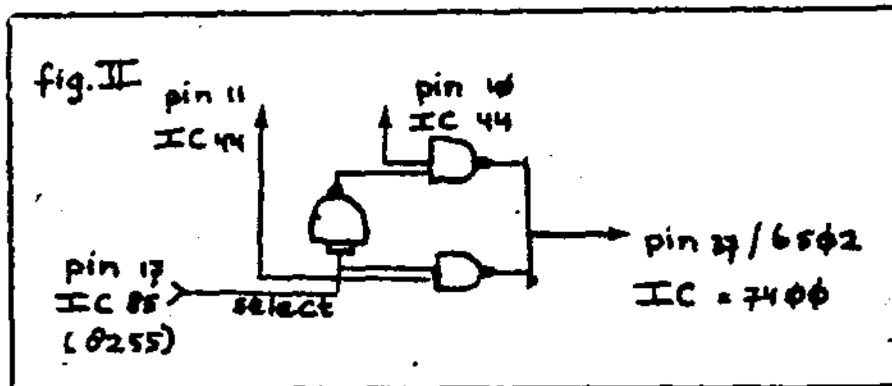


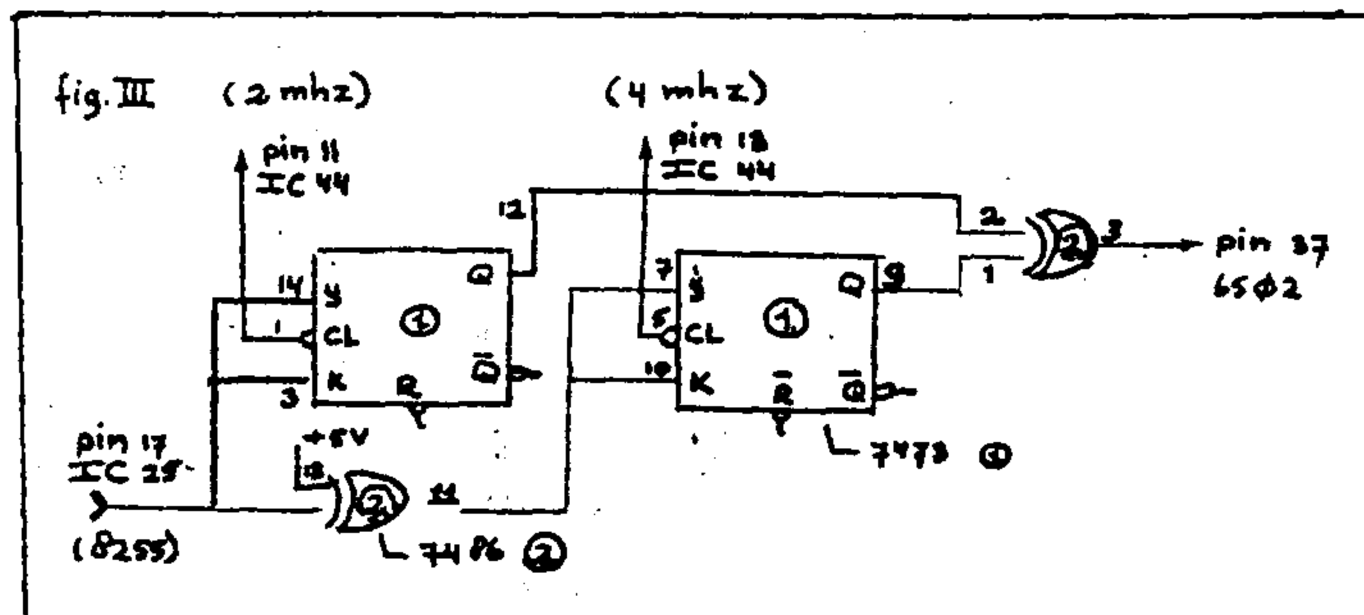
Quadruple 2 input
exclusive OR G

7486









"BUBBLE SORT"

```
1Ø REM"J.R.MARKS (16-8-'82)"
11 REM"OVERNAME VRIJ MET BRONVERMELDING."
12 P.$12'''"BUBBLE SORT."'';§=3
13 IN."NO. OF ELEMENTS"A;DIMBBA;B=#82ØØ;DIMZ65
14 IN."SORTING LEVEL"C;I.C<64;I=Ø;P."ELEMENTS:"';G.16
15 G.14
16 D.P.I" ";IN.$B;$B=$B
17 BBI=B;B=B+L.B+1;I=I+1;U.I>A O.$Z="SORT"
18 I=I-1;$BBI="";P."SORTING."'
19 F.J=C-1T.ØS.-1;D.E=Ø;F.K=ØT.I-1
2Ø I.J?BBK>J?BB(K+1);D=BBK;BBK=BB(K+1);BB(K+1)=D;E=1
21 N.;U.E=Ø;N.;P.$14''''''''''''''''"READY."''
22 F.J=1TOI;P.J" "$BBJ';N.;P.$15;LI.#FE94;RU.
```

toelichting:
het grote voordeel van dit programma is dat het niet de strings zelf verschuift,maar de pointers naar de string verzet.(tijdwinst)
deze pointers staan in de array bbi en de strings staan vanaf #82ØØ zo compact mogelijk.
"sorting level" is het aantal characters vanaf het begin van de string dat wordt bekeken.
dat levert dus een soort nauwkeurigheid van sorteren op.
als tijdens het invoeren van de strings blijkt dat het aantal te sorteren elementen te hoog is opgegeven voldoet het de string "sort" in te voeren om de invoer af te breken.

Hans M.,Groningen

monitor RH - deel 1

```
>
>
```

TOELICHTING oP de MONITOR.RH. (BandJe 6 in het Acorn Clubarchief)

ALGEMEEN: De monitor 9ebruikt de dubbele Punt, dus :, als PromPt. (In Plaats van de > in Basic). BiJ het verschiJnen van de : verwacht de monitor een oPdracht. Een oPdracht bestaat altiJd uit 1 letter of letterteken, dus b.v. een L of een M of een + etc. Wil men de oPdracht toePassen oP een adres, dan tyPt men na de : eerst dat adres in, 9evol9d door de oPdracht. Alles zonder sPaties. En het adres zonder # teken.
In de navol9ende toelichtin9 wordt AAAA 9ebruikt voor AANVANGSADRES en EEEE voor EINDADRES en DDDD voor BESTEMMINGSADRES (Destination).

V O O R B E E L D E N.

L De oPdracht L 9eeft 15 re9els machinecode-instructies vanaf het oP9e9even adres.(An9enomen natuurlijk, dat oP het oP9e9even adres een machinetaalPro9ramma staat, dat U wilt bekiJken en/of veranderen).

:AAAAL (returntoets).

<u>Voorbeeld:</u>

:C2B2L (returntoets)

```
C2B2  A9 29    LDA  @#29
C2B4  85 12    STA  #12
C2B6  A9 0D    LDA  @#0D
```

enz. 15 re9els.

[Voor de betekenis van de MNEMONICS LDA, STA enz. zie MANUAL Blz. 181 en verder.
Zie ook CURSUS ASSEMBLER in Acorn Nieuws. (Vanaf nr 6).
Zie ook ACORNSOFTPro9ramma 'PEEKO-COMPUTER'.]

De 15e re9el van het voor9aande voorbeeld van de L (List) oPdracht wordt:

```
C2CC  8D 21 03 STA  #0321
:
```

Door nu achter de : alleen L in te tyPen, vol9en de vol9ende 15 re9els. Dit kan men bliJven herhalen.

Conclusie: Het commando L 9eeft een disassembler, Pa9ina na Pa9ina.

* * * *

G Met de oPdracht G kan men naar een machinetaal Pro9ramma sPrin9en. OP dezelfde wiJze als dit 9ebeurt met LINK in Basic. En net als met LINK start ook G het machinetaalPro9ramma. Als voorbeeld kiezen we het machinetaalPro9ramma OSCRLF met als startadres #FFED. Zie Manual Hoofdstuk 25. Deze routine vol9t normaal oP het indrukken van de returntoets. Met het aanroePen van deze routine, 9evol9d door het indrukken van de returntoets, 9ebeurt dit dus TWEE maal; let oP !

Voorbeeld:

:FFEDG (returntoets)

Er zijn dus TWEE regels overgeslagen. Men noemt dit twee LINEFEEDS.

* * * *

P (P van Processor-registers).Met deze opdracht verschijnt de inhoud van alle processor-registers.

:P (returntoets)

Voorbeeld:

```
:P (returntoets)

 PC   A  X  Y  S  P  NV BDIZC
C2B4  29 00 05 FB 3D 00110101
```

(U kunt ook andere waarden krijgen).

Toelichting: De PC op de eerste regel betekent PROGRAM COUNTER. Na uitvoering staat de PC op C2B4. De A staat voor het ACCUMULATOR-REGISTER (normaal ACCU genoemd) en is geladen met #29.
X en Y zijn INDEX-REGISTERS van de 6502. De S staat voor STACK-POINTER. Een soort 'prikpen' met notities van door de 6502 te onthouden DATA of ADRESSEN.
De P staat voor PROCESSOR-STATUS-REGISTER. Hierin staan de beruchte FLAG's. Te weten N(Negatief), V(Overflow), B(BRK-flag), D(Decimale rekenwijze), I(Interrupt-blokkeerflag), Z(Als de Accu 0 wordt, dan wordt de Z-Flag 1), C(De Carry; de '1-onthouden' bij rekenen, U weet wel).

Waar dient deze opdracht voor? Wel; de waarden die U met de P opdracht afgedrukt krijgt zijn niet de werkelijke (momentele) inhouden van de registers, maar copieen die tevoren van deze registers gemaakt zijn.
U kunt deze copieen veranderen door NU een dubbele punt te typen en

vervolgens de waarden van de registers zoals U die hebben wilt. De eerste ingetypte waarde komt dan onder A, de tweede onder X enz. Een waarde die niet veanderd moet worden, dient dus opnieuw te worden ingetypt.Let ook op de spaties !

Voorbeeld:

```
:P (returntoets)

 PC   A  X  Y  S  P  NV BDIZC
3000  DA 03 A9 EA 31 00110001
::DA FF A0 D5 (returntoets)
:P (returntoets)

 PC   A  X  Y  S  P  NV BDIZC
3000  DA FF A0 D5 31 00110001
```

De laatste P-opdracht drukt dus het resultaat van de verandering af.
We hebben nu echter dus de c o p i e e n veranderd. Deze copieen worden in de echte registers teruggeschreven op het moment TUSSEN het geven van de opdrachts G, S en T en de uitvoering van deze opdrachts door de computer.

* * * *

S (van STEP) Met deze opdracht kan men 1 instructie doen uitvoeren van een machinetaalprogramma.

:AAAAS (returntoets).

Voorbeeld:

:C2B2S (returntoets)

```
C2B2  A9 29    LDA  #$29
 PC   A  X  Y  S  P  NV BDIZC
C2B4  29 00 05 FB 3D 00111101
```

(U kunt ook iets anders krijgen onder X Y en P)

Toelichting:
De eerste regel kennen we van de L opdracht, een disassemblerregel. De tweede en derde regel kennen we van de P opdracht. De eerste regel is de machinecode-instructie; de twee volgende regels geven de registers weer NA uitvoering van die eerste regel. U ziet, dat de PC is bijgesteld van B2 naar B4 omdat de instructie twee bytes lang was; de eerste byte A9 (de OPCODE:'laad ACCU met....) het tweede byte 29 (de OPERANT).

* * * *

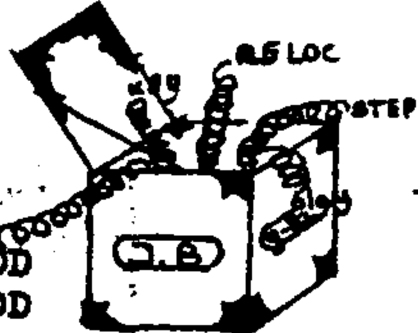
T (van TRACE) Met deze opdracht wandelen we door een machinetaalprogramma om het verloop na te gaan. Dit kan in de PAGINA-MODE wanneer Ctrl-N voor de opdracht wordt ingetoetst,gevolgd door returntoets.(Geeft geen ERROR). De opdracht T wordt beeindigd door ESC in te drukken en vast te houden tot een volle instructie is afgerond en de dubbele punt weer verschijnt. Let op: Bij twee keer ESC of dender gaat de monitor terug naar BASIC. De T opdracht is een automatisch herhalende STEP-opdracht.

* * * *

(wordt vervolgd)

ATOM ROM UITBREIDING

de "JOSBOX"



Commando's: READ DATA RESTORE FIND FCOS SCOS XDUMP HDUMP DISAS STEP PLAY SHAPE ON ERR RENUM COPY RELOC KEY GRMOD TXMOD

READ	FCOS	DISAS	ON ERR	KEY
DATA	SCOS	STEP	RENUM	GRMOD
RESTORE	XDUMP	PLAY	COPY	TXMOD
FIND	HDUMP	SHAPE	RELOC	

Alle commando's zijn af te korten zolang het maar één-duidig blijft; "X." voor XDUMP etc., FIND is niet af te korten, omdat we al "F."=FOR, "FI."=FIN en "FIN."=FINPUT hebben.
Het ROM zit op #AXXX. Een floatingpoint moet aanwezig zijn. Voor STEP moet een VIA en LINK 2 aanwezig zijn.
Er hoeft behalve FCOS en SCOS niets van te voren opgeroepen of "gelinkt" te worden, zoals bijv. bij de TOOLBOX.

1. READ, DATA, RESTORE.

```
Syntax: READ <VAR1>[,<VAR2>......,<VARn>]
        DATA [<ITEM1>,..........,<ITEMn>]
        RESTORE |
        RESTORE <LINE#>|
        RESTORE <LABEL>|
        RESTORE <EXPRESSION>
```

Alle drie zijn volledig geimplementeerd: er kunnen strings, integers en reals gelezen worden. DATA hoeft niet op het begin van een regel te staan.

VOORBEELD:

```
10 INPUT "INDEX:"I
20 RESTORE (10*I+50)
30 READ $TOP
40 PRINT $TOP'
50 GOTO 10
60 DATA "AAP"
70 DATA "NOOT"
80 DATA "MIES"
```

2. FIND.

```
Syntax: FIND "<STRING>"|
        FIND $<EXPRESSION>
```

FIND zoekt in een basic text space naar een opgegeven string en print de regel(s) uit, waarin deze string zich bevindt.

VOORBEELD:

```
FIND "0 TO 30"
$TOP= "TEST"
FIND $TOP
```

3. FCOS, SCOS.

FCOS zorgt voor 1200 baud tape transfer. De blokken worden aaneengesloten op de band gezet. (De gaps en de header zijn ingekort.)
Bij data transfer verkleurt de cursor (wit - grijs), dit gebeurt <u>niet</u> als: NOMON actief is of als
het scherm uit staat (P. $21) of als
GRMOD actief is (zie verder)

SCOS is idem aan FCOS alleen 300 baud.

TIMING:

	1K	4K	Bytes
FCOS	0.16	0.59	
SCOS	0.58	3.58	

Unnamed files gaan nog sneller, nog geen seconde gemeten.

4. XDUMP, HDUMP.

Syntax: XDUMP / HDUMP <START>[,[<EIND>]]
Wordt na de komma geen eindadres opgegeven, dan alleen te stoppen met ESCAPE.
Wordt [,[<EIND>]]niet opgegeven, dan wordt alleen 1 byte behandeld.
XDUMP geeft een gecombineerde ASCII / HEX dump.
HDUMP geeft alleen een HEX dump.

VOORBEELD:
```
XDUMP # A000, # AFFF
HDUMP # C2B2
```

5. DISAS.

Syntax: DISAS <START>[,[<EIND>]]
DISAS disassembleert het opgegeven geheugen stuk.
Wordt na de komma geen eindadres opgegeven, dan alleen te stoppen met ESCAPE.
Wordt [,[<EIND>]] niet opgegeven, dan wordt alleen 1 byte behandeld.

VOORBEELD:
```
DISAS # C2B2,
```

6. STEP.

Syntax: STEP <START>[,<DISPLAY>]
STEP single stept machine code vanaf adres <START>. Er wordt een disassembly gegeven van de instruktie met een lijst van de registers A, X, Y, PS, SP daarachter.
STEP wordt gestopt bij een BRK=# 00 of een RTI=# 40.
Normaal beëindigen met ESCAPE.
Wordt [,<DISPLAY>] opgegeven dan wordt boven beschreven listing pas gestart als DC=<DISPLAY>. (Handig bij saaie maar noodzakelijke lussen.)
A, X, Y kunnen gepreset worden d.m.v. de variabelen A, X en Y.

VOORBEELD:
```
A=#11; X=#22; Y=#33
STEP # C2B2
```

7. PLAY.

Syntax: PLAY <NOTE$_1$>[,<NOTE$_2$>,......,<NOTE$_n$>]
Speelt noten 3 oktaven
8 lengtes (+verlenging)
NOTE syntax: <FREQ> <LENGTH>|
<REST> <LENGTH>
<FREQ>= A/ B/ ...... G/ A'/ B'/ ...... G'/ A''/B''/ ...... G''
A#/ B#/ ..../ A'#/ .... / G''#
<REST>= R
<LENGTH>= 1/ 2/ 4/ 8/ 1./ 2./ 4./ 8.

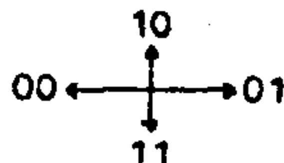
VOORBEELD: VADER JACOB
```
 10 REM VADER JACOB
 20 FOR X=1 TO 2
 30 PLAY C'4, D'4, E'4, C'4
 40 NEXT X
 50 FOR X=1 TO 2
 60 PLAY E'4, F'4, G'2
 70 NEXT X
 80 FOR X=1 TO 2
 90 PLAY G'8, A''8, G'8, F'8, E'4, C'4
100 NEXT X
110 FOR X=1 TO 2
120 PLAY C'4, G4, C'4, R4
130 NEXT X
140 END
```
P.S. E# = F, B# = C.

8. SHAPE.

Syntax: SHAPE <ADRESS>
SHAPE tekent een opgegeven figuur op het scherm volgens een "gepackte" SHAPE table op locatie <ADRESS>.
Het eerste byte van de SHAPE table geeft de lengte van de tabel. De volgende bytes geven informatie over de te plotten figuur. Als volgt:



De richting wordt als volgt bepaald:

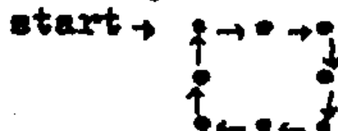


De plotparameter wordt als volgt bepaald:

- 00 : MOVE (onzichtbaar)
- 01 : SET (wit)
- 10 : INVERT (wit - zwart)
- 11 : BLANK (zwart)

Er komen dus 2 plot opdrachten per byte.

VOORBEELD:



MODE: SET

0101	0101	= #55
0111	0111	= #77
0100	0100	= #44
0110	0110	= #66

```
?#2800= 4; !#2801=#66447755
10 CLEAR 4
20 A= A.R.%256
30 B= A.R.%192
40 MOVE A,B
50 SHAPE #2800
60 GOTO 20
```

Het scherm wordt nu vol getekend met kleine vierkantjes.

9. ON ERR.

Syntax: ON ERR <BASIC STATEMENT>
Bij een fout wordt <BASIC STATEMENT>uitgevoerd, de FOR, DO, GOSUB stacks worden schoongemaakt.

VOORBEELD:

```
10 ON ERR P."ALLEEN GETALLEN!"'
20 INPUT "GEEF EEN GETAL "X
30 GOTO 20
```

10. RENUM.

Syntax: RENUM [<START>][,<INCREMENT>]
Hernummert een basic programma met alle GOTO's, GOSUB's en RESTORE's erbij.

VOORBEELD:

```
RENUM       (START=10,INCREMENT=10)
RENUM17     (START=17,INCREMENT=10)
RENUM,8     (START=10,INCREMENT= 8)
RENUM1,1    (START= 1,INCREMENT= 1)
```

Alle niet te hernummeren regels worden gelist.

VOORBEELD: GOTO (3*A+10)

11. COPY, RELOC.

Syntax: COPY / RELOC <START>,<END>,<DESTINATION>
COPY kopieert een stuk geheugen.
RELOC reloceert een stuk geheugen.

VOORBEELD:
```
COPY #A000,#AFFF,#2B00
RELOC #A000,#AFFF,#2A00
```
Bij RELOC worden alle adressen tussen <START> en <END> aangepast.

12. KEY.

Syntax: KEY <VARIABLE>
KEY doet een keyboardscan en kent de ASCII-waarde toe aan <VARIABLE>, anders wordt het 0.

VOORBEELD:
```
10 PRINT "DRUK ZO SNEL MOGELIJK "
20 PRINT "OP DE SPATIEBALK"'
30 T=0
40 DO
50 KEY K; T=T+1
60 UNTIL K=32 OR T=10
70 IF T<10 PRINT "OK"'
80 GOTO 10
```

13. GRMOD, TXMOD.

GRMOD vervangt de 6847 in de hoogste graphics resolutie ofwel normale tekst met tekeningen.
TXMOD schakelt GRMOD weer uit.

VOORBEELD:
```
10 GRMOD
20 PRINT "TEKENING 1"
30 FOR X=0 TO 180 STEP 3
40 MOVE X,0
50 DRAW 0,(180-X)
60 NEXT X
70 END
```

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

De lengte van de EPROM "JOSBOX" meet precies 4K machine code, er kan geen byte meer bij.

Er staat op de band:

AXXX #A000 #AFFF (2x)

ROM DEMO *

ELIZA *

ELIZADATA *

MUSIC *

* Behoeft geen uitleg.

Bij ELIZA: * RUN "ELIZA"

Alleen AXXX staat op 300 baud, voor de anderen is FCOS nodig.

Jos Horsmeyer



KURSUS MACHINETAAL

Door Leendert Bijnagte. Deel 2

*** De Stack pointer**

Dit is een 8-bits register. Dit register vormt samen met een vast getal (≠01) het eerste vrije stack adress. De stack wordt door de processor gebruikt en tijdelijke gegevens op te slaan.B.v. het terugkeer adres van een subroutine. Stack adress 01** - 01 FF

*** Assembler variable P**

De ruimte waar het programma moet worden geassmbleerd moet door de gebruiker worden bepaald. Moet het pal achter het programma komen dan wordt het DIM P-Dit wil zeggen dat het programma pal achter TOP geassembleerd wordt. Weet je van tevoren al een adres, dan geef je die waarde aan P. Dus b.v. P= ≠2800. Zorg dat bij een DIM- instruktie DIM P-1 de laatste instruktie is, omdat de ruimte(=lengte) niet vast ligt, die voor het programma nodig is. Met P legt U n.l. alleen het beginadres vast.

*** Haken**

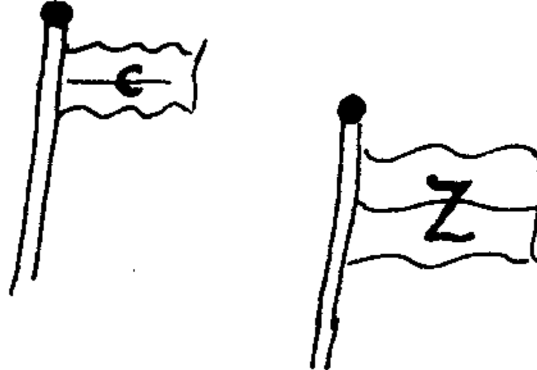
Het programma dat in machinetaal moet worden gezet,moet tussen haken worden gezet. Zo'n haak maakt de interpreter duidelijk, dat er een assembler programma tussen staat. Vergeet U de openingshaak, dan verschijnt (piep) ERROR 94 in line XX. Bij vergeten eindhaak ERROR 208 in line XX.
Een programma ziet er dus minimaal zo uit.

```
10   P= ≠2800     of DIM P-1
20   [
--
--
100 ]
110  END
```

***** Vlaggetjes dag*****

Het status of vlaggenregister is een belangrijk onderdeel van de 6502.
Een vlag is een hulpmiddel voor de programmeur. Een vlag kan 1 Of 0 zijn.
Als de processor daartoe opdrachr krijgt, kijkt hij of de genoemde vlag 0 of 1 is. U kunt hem vergelijken met de IF- THEN. Als de voorwaarde achter IF waar is(1 is dus) THEN...... Na iedere instruktie wordt het statusregister bijgewerkt.

* Indeling Statusregister *



N/V/-/B/D/I/Z/C/

De vlaggen stuk voor stuk:

C — De carry vlag

Deze vlag wordt gezet als bv. bij een optelling de ≠FF wordt overschreden. U kent hem vast wel van de lagere school: één onthouden als de optelling groter is dan 9.Bij aftrekken is het net andersom. De carry wordt 0 als er "geleend" wordt;bv. 10014-9=10005. Bij schuiven en roteren kan de carry worden gebruikt als 9e bit.(Hierover volgt een aparte aflevering.)

Z — De Zero vlag

Deze vlag wordt gezet (1) als de voorafgaande bewerking op 0 uitkomt. Dus bij 5-5=0 wordt de zero vlag gezet.

I — Interrupt bit

Dit is eigelijk een apart verhaal.De processor bezit een ingang IRQ.Als deze ingang 1 wordt, kan de processor, afhankelijk van de stand van de vlag,gedwongen worden om een ander stukje programma uit te gaan voeren. U kunt de vlag vergelijken met een direkteur: Als de vlag 1 is hangt het bordje "niet storen" op de deur. Als de vlag 0 is mag er wel gestoord worden. Het storen gebeurt niet direct. De processor maakt eerst zijn instructie af en kijk dan naar de interrupt.($t_{max.}$ =3microsec.)

D — Decimaal/Hex vlag

Als deze vlag gezet is, zullen op- en aftellingen decimaal gebeuren. Is de vlag nul dan zullen alle rekenkundige bewerkingen hexadecimaal geschieden.

B — Break vlag

Als de processor tijdens een programma een BRK-instrc. tegenkomt dan wordt deze vlag gezet en springt hij via de brk-vector weg.Wordt ook wel software interrupt genoemd.

– — Bit 5

Wordt niet gebruikt.

V — Overflow vlag

Deze vlag wordt gezet als een bewerking groter is dan 64 (≠3F).

N — Negative vlag

Deze vlag wordt gezet als het MSB van een getal 1 is; groter dan 128 (≠7F). Als bv. een getal wordt afgetrokken van een getal en de N-vlag wordt hierbij gezet,dan is het getal dus negatief.

Sommige vlaggen kunnen ook door de programmeur worden "geset" of "reset":

CLC=clear carry (maak de carry vlag 0)	SEC=set carry(maak de carry vlag 1)
CLD=clear decimaal(maak de D-vlag 0)	SED=set deci.(maak de D-vlag 1)
CLI=clear interrupt(maak de I-vlag0)	SEI=set inter.(maak deI-vlag 1)
CLV=clear overflow(maak de O-vlag 0)	

Afhankelijk of een vlag 0 of 1 is kan er gesprongen worden naar een ander adres. We hebben de volgende mogelijkheden:

BCC=spring als de C-vlag 0 is.	BVC=spring als de V-vlag 0 is.
BCS=spring als de C-vlag 1 is.	BVS=spring als de V-vlag 1 is.
BNE=spring als de Z-vlag 0 is.	BPL=spring als de N-vlag 0 is.
BEQ=spring als de Z-vlag 1 is.	BMI=spring als de N.vlag 1 is.

DE SCHAKELKAART

In samenwerking met onze Programmeurs Foppe Bouma en Arie Marchal werd door Roel Heuvel en ondergetekende een 'ideale' schakelkaart ontwikkeld voor de Acorn Atom.

Eigenschappen: De kaart decodeert en gebruikt één schakelbit uit de grote Atom-voorraad, n.l. #Bfff. "Soft" - schakelen gebeurt door dit adres te vullen met een getal 0 tot 16.

De kaart bevat 10 stuks 24 pin sockets voor ieder 4 k. Chips. Dit kunnen zijn: Toolboxen, Editors, Calculators e.d. in ongewijzigde handelsvorm, dus geadresseerd op A000.

Van deze 10 sockets zijn er 4 geschikt voor plaatsing van ieder 4k CMOS geheugens (gestapelde 2k's), waarvan 2 geadresseerd naar E000en 2 naar A000.

De kaart komt schakeltechnisch 'buiten de bus' en is geheel opnieuw gebufferd. Op het moederboard moet dus A000 buiten de bus gebracht worden. De hierdoor vrijkomende socket IC24 kan dan gebruikt worden voor een 2e 4k. op DXXX. De 'ombouw' van het moederboard vraagt 3 diodes en enige vakkennis.

De print + minimaal benodigde onderdelen zal ongeveer ƒ 125.- gaan kosten. Voor gebruik binnen zowel als buiten de kast is voorzien in 2 connectors, male/female.

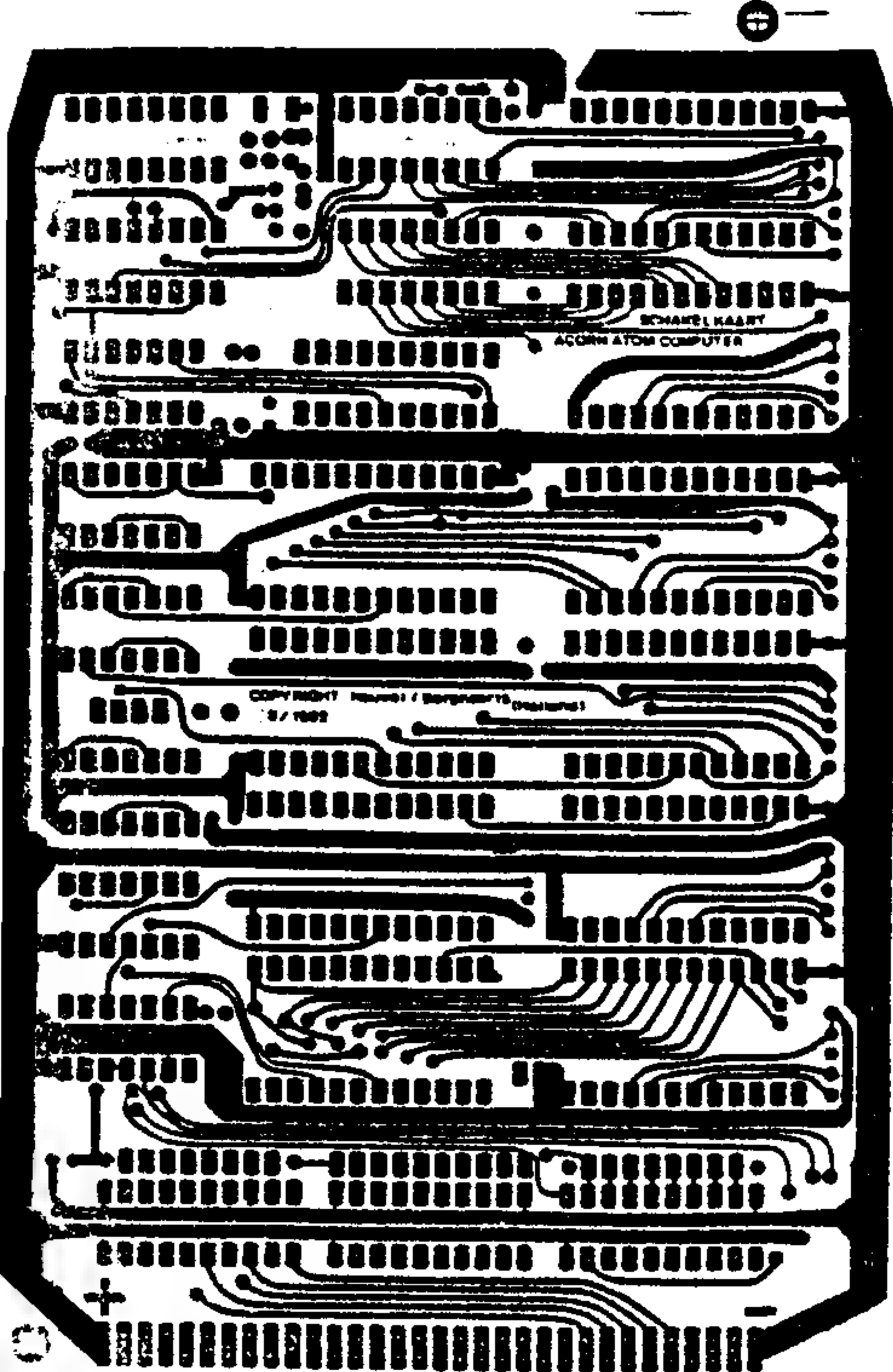
De kaart is in principe ontwikkeld voor eigen gebruik door de kern van de wat meer ervaren leden. Ze kan vrij van copyright ter beschikking gesteld worden voor alle clubleden. Aangenomen echter, dat de gehele zorg voor verspreiding, verrekening, bouwhulp e.d. binnen de REGIO georganiseerd wordt. Wij nemen aan, dat U begrip zult hebben voor deze noodzaak.

Samenvatting: In eenvoudigste vorm is de kaart geschikt voor het softschakelen van (handels)-utility's op A000 tot 6 stuks en vraagt het 'buiten de bus' brengen van het adres A000-AFFF.

Door bijplaatsen van 2k. CMOS IC's wordt het mogelijk om op AXXX en EXXX 'eigen' routines te plaatsen en in prgramma's te gebruiken zowel als voor het onderbrengen van SPRONG-routines (Jumpers) naar elke denkbare geheugenplaats van de ATOM, óók naar de geheugenkaart uiteraard.

Zie hiervoor de bijdragen van Foppe Bouma. Mét het gebruik van EXXX komen voor de hobbyist uiteraard ook de +DOS commando's vrij ter beschikking. Schakeltechnisch blijft er met deze kaar niet veel meer te wensen over. Een belangrijk aspect is, dat het de hobbyist mogelijk wordt 'eigen' routines uit te proberen en tot toolboxen samen te stellen.

Daarvoorgeschikte compacte routines hiertoe worden ingezameld en gepubliceerd. Zie b.v. 'Cataloog'(Marchal), 'verhuizer', 'Viscos', 'Sort(corder)', 'Renumber' 'Fast Cos', 'Monitor AP' enz.enz.


SCHAKELKAART
ACORN ATOM COMPUTER
COPYRIGHT

OVER SOCKET IC 24 en SCHAKELKAARTEN

(en d u s over de hele ATOM)

Van de 64 k.bytes aan geheugenruimte, die de ATOM (en elke 6502 computer) ter beschikking heeft, zijn er 4 k. (rond 4100 stuks), geadresseerd van ≠ A000- ≠AFFF en te bereiken aan de 'vrije' socket van IC 24.- In die socket kun je handelsverkrijgbare tool-boxen en editors duwen en zoiets stond de ontwerpers van de ATOM waarschijnlijk ook voor ogen. Een socket voor "UTILITY's" dus.

Gebruikt men méér dan een zo'n utility, dan kan men deze beurtelings in de socket duwen en weer uittrekken.

Aangezien echter slechts één pin van de 24 stuks bepaalt of een 4 k. Eprom áánstaat of uitstaat, kan men deze Utility's ook gemakshalve op elkaar stapelen als 'varkens' (zie A.N. nr 3 blz 8) en alleen hun 'aan-uit' pin, de CHIP-SELECT pin omschakelbaar maken, (zie A.N.4 blz 17) met een stappenschakelaar b.v.

Wil je het netter doen, dan kun je een printje maken met meer sockets en dit printje met een zgn WIRE-WRAP socket in de vrije socket steken (waarbij de vrije socket de vernieling ingaat) of deze met een stukje flatkabel aan de vrije socket verbinden.

U kunt zo'n printje zelf maken volgens het schakelschema van ARIE MARCHAL (Bibliotheek nr. 16, 1blz.). U kunt zo'n printje kopen voor ƒ 79.- waarop 4 UTILITY's kunnen (en ook moéten, anders loopt de zaak heet). Of voor ƒ 179.- een zgn. 'wisselrack'; een beetje bombastische doos waar zegge en schrijve 2 utility's in kunnen, ieder midden op een print.

Tot zover hadden we het over 'met de hand' omschakelen; met een mechanische schakelaar. <u>HARDWARE-OMSCHAKELEN</u> dus. Met de schakelaar kiezen we in feite alleen de CHIP-SELECTLIJNEN. De andere 23 pinnen zijn parallel geschakeld. D ó ó r g e l u s t heet dat.

Het omschakelen kan echter ook <u>SOFTWARE-MATIG.</u> Hierover het volgende: Om software- om te schakelen heeft men minstens 1 zgn. SCHAKELBITJE nodig. Dat is een bitje, een flipflop, die het vermogen heeft om een externe lijn hóóg te houden of láág te trekken, dus 5 Volt te houden of 0 Volt te maken. Gewone geheugenbits hebben daarvoor niet het vereiste vermogen. Zulke schakelbits zitten in DECODERS (waarover later) in LATCHes (idem) en in speciale 'INTERFACE-ADAPTERS' zoals bij de ATOM in de VIA 6522 en de 8255.

Op het moederboard heeft de 8255 één zo'n schakelbitje over, PC3. We gebruiken deze (o.a.) om de printer naar GRAFICs te schakelen. Zie A.N. nr.5 blz 25. De 6522 heeft twee machtige, gehele poorten van ieder 11 stuks aan schakelbits. Eén zo'n poort (poort A) stuurt de printer uit met DATA en CONTROL-lijnen, de zgn. HANDSHAKE (dat betekent zoiets als : Nu Jij, Dan Ik enz.).

HET IS ZEER VERLEIDELIJK OM VOOR ALLERLEI 'SOFT-OMSCHAKELDOELEINDEN' DE BITJES VAN DE B-POORT van IC 6522 TE GEBRUIKEN. WIJ STAAN HIER ECHTER ZEER GERESERVEERD TEGENOVER. Waarom ?

Wel; de B-Poort van IC 6522 is een volledige machtige poort voor c o m p l e t e schakeldoeleinden. Van zeer veel leden van onze club weten wij voor welke doeleinden deze complete poort reeds gebruikt wordt, varierend van het sturen van de wachmachine, de beveiligings-installatie, enz tot de prisé-sterrenkijker aan toe.

Deze leden, die de gehele poort gebruiken voor doelen, waarvoor deze is ontworpen, kunnen wij niet aandoen om te propageren om u i t die poort losse bitjes te plukken voor allerlei geschakel elders. Wij vinden dit geknutsel, dat onherroepelijk binnen afzienbare tijd moet vastlopen. Bovendien absoluut overbodig omdat de ATOM mogelijkheden in overvloed heeft om elders SCHAKELBITS aan te hangen. Volgens de globale berekening op blz 6 van A.N. nr.3 zelfs 1300 stuks.

Wij staan derhalve zeer critisch tegenover 'schakel' kaarten zowel van leden als vanuit de handel, die in feite geen schakelkaarten zijn, maar printen met doorgeluste sockets, en voor het schakelen de B-Poort gebruiken. Door het aureooltje van 'softomschakelbare' kaarten (handelskaarten) kijken wij heen en trappen er niet in. Voor de 6522, die in feite het schakelwerk doet, hébben we al een keer betaald.

OM derhalve verdere misverstanden te voorkomen noemen wij in ACORN NIEUWS ALLEEN DIE KAARTEN 'SOFT-OMSCHAKELBAAR' die de benodigde schakelingen bevatten om uit de reserve voorraad van de ATOM een schakelbitje te kiezen en deze voor het soft-omschakelen te gebruiken. En de B-Poort met rust laten! Een dergelijke schakelkaart is, voorzover wij weten, niet in de handel en daarom maakten wij (R.Heuvel/G.Borghaerts) er zelf een.

= = =

Er is echter nog een tweede reden, waarom wij critisch kijken naar schakelkaarten voor de ATOM die van elders komen.

Dit heeft te maken met de visie die wij allengs hebben verkregen met betrekking tot de meest doelmatige verdere hardware/software ontwikkeling van de ATOM. Wie een product koopt, zit aan dat product vast. En dit vastzitten kan een ontwikkeling stimuleren maar ook danig afremmen.

Een voorbeeld kan dit verduidelijken. Kijken we naar de Floating Point. Voor velen is de Floating Point een prachtig en onmisbaar stuk computergereedschap. Maar zeker niet voor iedereen. Gemiddeld genomen is het nut van de Floating Point zéér betrekkelijk terwijl deze EP(ROM) eens en vooral 4k. van ons geheugen inpikt en het als een'must' wordt gezien om zo'n ding te kopen. Voor minder geld zet je ergens 'soft-omschakelbaar' in de ATOM en speciale rekenchip van Hewlett Packard die heel wat meer presteert en we hebben onze 4 k voor (ook) andere doeleinden terug. De afgelopen maanden, -en vooral weken- is er door de hard- en software top van de club geanaliseerd en (nachtenlang) gediscusseerd, ook over die Floating Point. Als conclusie zouden wij deze (EP)ROM het liefst (figuurlijk gesproken) uit z'n socket trekken en ergens inschakelbaar wegzetten voor wie hem gebruikt.

We stuiten daarbij op een gegeven dat wij niet missen kunnen, n.l. dat de Floating Point in z'n huidige vorm alle statements en instructies, die de ATOM t/m de F.P. niet begrijpt, voor onderzoek doorstuurt naar de adresruimte #D000, de lege socket van IC 24.
Dat kan echter ook anders. Door slechts 4 bytes van de 4100 in de F.P. te wijzigen, stuurt deze F.P. door naar een ander adres.
De F.P. in mijn ATOM stuurt door naar # EXEX.

Dit doorsturen naar ## EXXX is bedoeld om uit te vissen hoe programmeerbaar utility's worden, wanneer je £XXX als centrale 'draaischijf' van de ATOM gaat gebruiken. Onze top-programmeurs zijn hiermede reeds bezig en het lijkt het ei van Columbus. Een soft-schakelkaart op IC 24 met utility's dient rekening te kunnen houden met 'aansturen' vanuit meerdere adressen. In onze schakelkaart kan dat.

Dit betreden van de EXXX geheugenruimte zal hier en daar, vooral vanuit de handel, op verzet stuiten. Met EXXX is het al net ongeveer zo als met de F.P. ruimte. Het wordt voorgeschoteld als een 'must' om daar een Disk Operating System aan te hangen voor duizenden guldens. Nu is een disk-systeem best lollig om daar een 'Back-Up' van al je programma's neer te zetten; maar voor een beetje Disk-Operating is de ATOM ongeschikt. Wie van de leden werkelijk voor z'n hobby een Disk nodig heeft mag het mij berichten en wie mij kan voorrekenen dat al het kunst en vliegwerk aan de ATOM voor een goedlopend administratief systeem goedkoper is, dan het direct maar aanschaffen van een Pearcom, moge dit proberen.
Kortom: de 'must' om voor een Disk blijvend 4k. gereserveerd te houden wordt door ons niet gezien. Wij menen, dat je met deze geheugenruimte -ook- iets anders moet kunnen doen. De DRAAISCHIJF b.v.

Het mooiste is natuurlijk, wanneer die ruimte EXXX met de CMOS geheugen-IC-s (naar wens) bezet kan worden. Zodat deze DRAAISCHIJF continu toegankelijk is voor uitproberen, wijzigen en op slot doen.
Een schakelkaart die dat kan is uiteraard niet in de handel. Daarom maakten wij die voorziening op onze schakelprint.

═══════

Gaan we nu terug naar de Socket IC 24. Zit daarin een 'soft-schakelbare' print voor meerdere Utility's, dan kunt U -dit als voorbeeld- intypen: REN. De basicROM begrijpt dit niet en stuurt door naar de F.P. Deze begrijpt het ook niet en stuurt door naar EXXX. Op EXXX staat een programma'tje (van Foppe Bouma b.v.) die dat wel begrijpt; vervolgens op de Utility-kaart op b.v. socket 5 de ProgramPower Toolbox selecteert en daaruit het program REN. initieert. U kunt dan Renumberen.

= = = =

Daarmee zijn we er nog niet. Van de 4k. handelsboxen zijn er vele te koop. De ene heeft dié combinatie; de andere weer een andere. Als je iets specifieks nodig hebt koop je wel de hele box. En er is er géén, waar een goed sorteerdertje in zit b.v.
Kortom: het zou mooi zijn, als de schakelkaart óók de mogelijkheid had om op A000 '4k CMOS' geheugen te schakelen zodat we daarin onze eigen toolbox kunnen samenstellen met alleen dié routines die men nodig heeft en gebruikt. Heb je daarmee 4k vol, dan schrijf je die in EPROM en schuift deze door naar een andere socket, zodat de CMOS weer vrij komt. Wij brachten die mogelijkheid aan op onze kaart.

En dan zou het ook nog mooi zijn, wanneer je als EPROM zowel de 2732 als de 2532 als de 2716 zou kunnen gebruiken. Nou; ook dat kan.

= = = = = = = = = = = = = =

Wij vertellen U dit verhaal niet om U uit te nodigen onze schakelkaart te bestellen. Voorlopig leveren wij deze kaart alleen maar aan de beperkte "oude korn" van de Club en aan de Softwarespecialisten van de REGIONALE CLUBS om de zaak eens goed uit te proberen. En vervolgens de levering en nazorg op zich te nemen van verdere verspreiding binnen de regio.

HET GAAT IN DIT VERHAAL IN HOOFDZAAK OM DE VISIE DIE WIJ HEBBEN OP DE VERDERE ONTWIKKELING VAN DE A T O M ALS 'DOE HET ZELF' COMPUTER. ALLE PRODUCTEN DIE WIJ TEGENKOMEN VERGELIJKEN WE MINSTENS MET DEZE VISIE.
Constructieve bijdragen van leden t.a.v. het voorgaande :Zeer welkom !



Atom EPROM programmer

THE ACORN ATOM has a vacant expansion socket inside for a utility ROM which will accommodate up to a 4K by eight-bit device, writes John Flower of Cowplain, Hampshire. The EPROM which can be fitted is the 2532, which is made by Texas Instruments and a number of Japanese suppliers. This device is a 32K chip organised as 4,000 words of eight bits. The chip is readily available for about £6.

You can program your favourite game or a useful machine-code routine on to this device so that it can become a permanent part of your computer's operating system. You could even write your own toolkit program. The 2532 EPROM is remarkably easy to program with the Atom since most of the necessary circuitry for an EPROM programmer is already inside the Atom's VIA chip. If you decide to have a go at building this design then you will have to buy and fit the optional 6522 versatile interface adaptor chip to your machine.

For a programmer interface you only need to provide an address latch externally to address each location in turn while the data is presented and the programming operation takes place. The circuitry consists of two LS374 eight-bit latches to hold the address word plus a zero-insertion-force socket to carry the EPROM without risk of bending the pins.

Pin 21 of the 2532 EPROM is connected to +5 volts when reading data. Data is read by pulsing pin 20 low while looking at the data pins. The device works in reverse if pin 21, which is normally held to 5 volts, is taken to 25 volts. In this case data present at the data pins is programmed into the eight locations whose address is present at the address pins. This happens if pin 20 is pulsed low for exactly 50ms.

The program performs the necessary operations to copy, program and verify EPROMs. Programs to be copied from an EPROM or to be programmed into one are stored in the graphics memory from location #6400 onwards. The program is menu driven and prompts to see what operations you wish to perform.

The VIA chip writes the relevant address on to ports A and B and clocks the address latch to store the 12-bit address. Then the eight-bit data is either presented to or read from port B.

Construction of the circuit should present little difficulty. Only two integrated circuits are involved, plus four resistors, two diodes, an npn transistor and one capacitor. You will need to fit an Acorn bus connector to your computer and buy the appropriate 64-way Eurocard DIN connector mating socket.

The circuit diagram shows the bus connections that are used, viewed from the rear of the programmer card. The 64-way right-angle plug is fitted to the Atom and the socket is fitted to a piece of Veroboard upon which the programmer is to be constructed.

The light-emitting diode serves to show that Read or Write operations are taking place. The other diode — 1N914 or similar — ensures that +5V is connected to pin 21 of the EPROM when normal reading of the device occurs. When +25V is applied to the device for programming purposes, the diode becomes reverse-biased so that the 5V power supply is not affected. It is very important to connect this diode the right way round to avoid serious damage to the computer's power supply. The anode of the diode must be connected to the Atom's +5V supply. The cathode is usually marked by a thick painted ring on the diode body and should be connected to pin 21 and the programming switch.

After constructing the circuit you do not need the programmer card to prove that the program is working: with the card unplugged the program will com-

(continued on page 156)

EPROM programmer card circuit.

(continued on page 152)

#FF into each screen RAM location. You can check this by first clearing the mode 4 screen, then adding line 95

?#B000 = #F0

Running the EPROM copy routine will let you see the memory being filled, which takes about four minutes. The verify routine reads the EPROM and compares it with the contents of the screen RAM. Any errors will be listed and a simple check-sum is computed by adding the decimal contents of each location together. See if the answer for the all-#FF case comes to 4,096 × 255.

Having satisfied yourself that the program works plug in the programmer card and set the Read/Program switch to Read, connection to +25V open-circuit. The diode has been connected correctly if you can measure 5V on pin 21 of the ZIF socket.

The light-emitting diode should flash during the Copy, Verify and Program modes. If it does not, it may be connected round the wrong way, so try reversing it. Next insert line 95 to watch the memory being loaded with the copy routine. If you short-circuit pin 20 to each data pin in turn on the ZIF socket, you can see from the bit pattern whether each data bit is being read. Do not try to fit an EPROM or connect 25V until you are satisfied that everything is in order.

Try reading an EPROM or ROM. You could unplug and read the floating-point ROM if you have a fully expanded Atoms. When you are ready to program a 2532 EPROM you can connect a 25V supply to the programmer card: four 6V transistor radio batteries connected in series are a simple way of obtaining it.

If you follow the program instructions to the letter, there should be no problem in programming your own EPROMs, and when you can program EPROMs, you will soon want to erase them. Unfortunately there is no simple method of doing so: intense ultraviolet radiation at a wavelength of 275.8nm, is required.

You can buy small fluorescent tubes which radiate at the correct frequency, but if you make your own eraser it is essential to mount the tube in a light-tight box to protect the eyes from harmful UV radiation. EPROMs to be erased should be exposed for about 20 minutes about 3cm. from the tube. Commercial erasers usually have an automatic timer and safety cut-out switch to disconnect the light if the box is opened. Be careful with the erase timing since too much UV bombardment can shorten the number of program-erase cycles obtainable from the EPROM.

Programmer card component list.

Component	Quantity	Comments
74LS374	2	eight-bit latch
BC108	1	transistor; any npn switcher will do)
1N914	1	diode (any 50V PIV diode suitable)
LED	1	any type
4,700ohm resistor	3	
470ohm resistor	1	may be omitted if diode LED not fitted
Single-pole switch	1	
64-way Eurocard DIN connectors (plug and socket)		
Verocard or similar prototyping card, about 5in. square		
24-pin zero-insertion-force socket		

UV lamps can be obtained from Watford Electronics, 33-35 Cardiff Road, Watford, Hertfordshire.

EPROM programmer.

```
10DIMT(1);@=5;?#208=#55;?#B80C=#CC
15P.$12"    2532 eProm Programmer"'''
20REM: J C FLOWER 1981
25P."PROGRAMS MUST RESIDE IN"'
30P."4K BYTES STARTING AT #8400"'
35P."DO YOU WISH TO PROGRAM (P)"'
40P."COPY (C) OR VERIFY (V)"''';P."enter letter"''';INPUT$T
45 IF$T="P" GOS.a
50 IF$T="C" GOS.b
55 IF$T="V" GOS.c
60 G.15
65bP.$12;P.''"    eProm copying facility"''
70P."check THAT READ/PROGRAM SWITCH"'"IS ON read"''
75P."INSERT EPROM WITH PIN 1 ADJACENT"'"TO ZIF SOCKET LEVER."
80P.'''"WHEN READY, Press a letter key"
85LINK#FFE3;P.$12'''''
90P."          "$128"reading"$128"eProm"$128'''
95B=0;A=0;?#B80C=#CC;C=#8400;Z=0;S=0;?#E1=0
100DO
105?#B80E=#00
110?#B802=#FF;?#B803=#FF
115?#B800=A;?#B801=B
120?#B80C=#CE;?#B80C=#CC
125?#B802=#00
130?#B80C=#EC;?C=?#B800;S=S+?#B800
135?#B80C=#CC;C=C+1
140P.(B+Z)'$11;B=B+1
145IF B=256 A=A+1;B=0;Z=A*256
150UNTIL B=0 AND A=16
155P.'''"checksum="S'"PRESS A KEY";LINK#FFE3
160R.
165aP.$12;P.'"    eProm Programming facility"''
170P."check THAT READ/PROGRAM SWITCH"'"IS ON read"''
175P."EPROM WILL BE PROGAMMED WITH 4K"'"BYTES STARTING AT LOC."
180P." #8400"''"INSERT EPROM WITH PIN 1 ADJACENT"'
185P."TO ZIF SOCKET LEVER."''
190P."WHEN READY, SWITCH READ/PROGRAM"'"SWITCH TO Program"'
195P."Press"$128"a"$128"letter"$128"key";LINK#FFE3
200P.$12'''''"          "$128"Programming"$128"eProm"$128'''
205B=0;A=0;C=#8400;Z=0;?#E1=0
210DO
215?#B80E=#00
220?#B802=#FF;?#B803=#FF
225?#B800=A;?#B801=B
230?#B80C=#CE;?#B80C=#CC
235?#B800=?C;?#B80C=#EC;WAIT;WAIT;WAIT;?#B80C=#CC;C=C+1
240P.(B+Z)'$11;B=B+1
245IFB=256 A=A+1;B=0;Z=A*256
250UNTIL B=0 AND A=16
255P.$7$7$7$7$7$7$7$7$7$7$12''''"SWITCH READ/PROGRAM SWITCH"'
260P." TO read"''"Press a letter key";LINK#FFE3
265R.
270cP.$12;P.'"    eProm verifying facility"''
275P."check THAT READ/PROGRAM SWITCH"'"IS ON read"''
280P."EPROM CONTENTS ARE COMPARED WITHTHE CONTENTS OF"
285P." MEMORY.STARTING AT LOC.#8400"'"ERRORS WILL LIST"
290P.'''"Press"$128"a"$128"letter"$128"key";LINK#FFE3
295P.$12'''''"          "$128"verifying"$128"eProm"$128'''
300B=0;A=0;C=#8400;Z=0;S=0;?#E1=0
305DO
310?#B80E=#00
315?#B802=#FF;?#B803=#FF
320?#B800=A;?#B801=B
325?#B80C=#CE;?#B80C=#CC
330?#B802=#00
335?#B80C=#EC;V=?#B800
340?#B80C=#CC
345P.(B+Z)'$11
350IF V=?C G.360
355P."ERROR AT "(B+Z),&V'
360C=C+1;B=B+1;S=S+V
365IF B=256 A=A+1;B=0;Z=A*256
370UNTIL B=0 AND A=16
375P.'"checksum",S'"PRESS A KEY";LINK#FFE3
380R.
```

J.H. Schoon
Lonnweg 149
3137 NH Vlaardingen

Geachte heer Borghaerts,

In Acorn Nieuws nr. 6 las ik o.a. over de mogelijkheden de Acorn uit te breiden met 16 k RAM in diverse uitvoeringen. Diverse leden zouden daarbij met goed resultaat en naar tevredenheid gebruik hebben gemaakt van de door Elektuur ontwikkelde en gepubliceerde 16 k dynamische RAM print.

Ook ik heb dat gedaan, zij het aanvankelijk met grote problemen.. Ik heb de originele Elektuur print gebruikt en exact de door hen aangegeven onderdelen. De zaak bleek echter niet helemaal juist te functioneren. Bij het testen met verschillende RAM-test programma's bleken willekeurige adressen op willekeurige momenten zonder meer te worden gewijzigd qua inhoud. U begrijpt de ene foutmelding na de andere. Nog verder ontstoren van de voeding, het plaatsen van een netfilter in de 220 Volt hielp allemaal niet.

Bij telefonisch contact tijdens het bekende vragenuurtje met de redactie van Elektuur werd mij gezegd dat het probleem bekend was en meer voor kwam. Men had al een oplossing op papier staan. Ik moest echter het probleem wel schriftelijk indienen (waarvoor is dan het vragenuurtje ?). Uiteindelijk kreeg ik na ca. 4 weken een kopie van het wijzigingsadvies, dat ik direct heb uitgevoerd op de print. Kortweg kwam het erop neer, dat enkele printsporen d.m.v. extra draadbruggen "verzwaard" moesten worden; C1 werd een trimmer van 80 pF; C3 vervalt en C5 wordt 220 pF. M.a.w. de problemen moeten worden gezocht in een niet goed functioneren van de refreshment logica.

Welnu, na het uitvoeren van deze wijzigingen bleek het middel nog erger dan de kwaal geweest te zijn. Op vrijwel alle adressen nu foutmeldingen.

Na enige dagen experimenteren met condensatoren etc. is het dan toch gelukt om de zaak goed werkend te krijgen. Wel is het nu zo dat de dimensionering van een aantal onderdelen bij mij nu heel anders is dan Elektuur aanvankelijk publiceerde resp. in het wijzigingsblad voorstelde.

Mochten er leden zijn die ook met problemen zitten t.a.v. deze print, dan kunnen zij altijd contact met mij opnemen en zal ik hen graag adviseren hieromtrent.

Zoals u mogelijk weet wordt door zendamateurs gebruikt gemaakt van een zg. "QTH-locator", een "code" van 3 letters en 2 cijfers, waarmee op enkele kilometers nauwkeurig de locatie wordt aangeduid vanwaar men zendt. Hierdoor is het m.b.v. een grootcirkelberekening mogelijk om vrij nauwkeurig de afstand tussen beide stations te bepalen. Dat het met een computer wel heel erg eenvoudig is, hoef ik daarbij niet te vertellen.

Een listing van het door mij voor de Acorn geschikt gemaakte programma doe ik u hierbij toekomen.

Met vriendelijke groeten

Hans Schoon.

Afstandsberekening.

Dit programma berekent de afstand in kilometers tussen 2 plaatsen op de aardbol, welke worden aangeduid met de door zendamateurs gebruikte QTH-locator.

```
 10 P.$12,"AFSTANDSBEREKENING"''
 20 DIM A(5)
 30 IN."TUSSEN QTH-LOCATOR"$A
 40 P."EN: "'
 50 GOS.a
 60 %B=%D
 70 %C=%E
 80dIN.'"QTH-LOCATOR"$A
 90 IF $A="" G.z
100 GOS.a
110 %F=%D
120 %G=%E
130 %H=57.2957795
140 %I=%B/%H
150 %J=%F/%H
160 %K=%C/%H
170 %L=%G/%H
180 %M=SIN%K*SIN%L
190 %N=COS%K*COS%L*COS(%J-%I)
200 %O=%M+%N
210 %P=(SQR(1-%O*%O))/%O
220 %Q=ATN%P*111.11*%H
230 £=20:P.ABS(%Q)," KM"'
240 G.d
250aR=A?0
260 S=A?1
270 T=A?2
280 U=A?3
290 V=A?4
300 IF R<85 R=2*R-130:G.b
310 R=2*R-182
320bIF S<86 S=S-25:G.c
330 S=S-51
340cT=T-48
350 U=U-48
355 IF U=0 U=10
360 IF V=65 Y=5:Z=3
370 IF V=66 Y=5:Z=1
380 IF V=67 Y=3:Z=1
390 IF V=68 Y=1:Z=1
400 IF V=69 Y=1:Z=3
410 IF V=70 Y=1:Z=5
420 IF V=71 Y=3:Z=5
430 IF V=72 Y=5:Z=5
440 IF V=74 Y=3:Z=3
450 %D=-(U/5+R-Z/30)
460 X=7:IF U=10 X=8
470 %E=(X-T)/8+Y/48+S
480 R.
490zP.$12,"EINDE"';E.
```

N.B. voor "apestaart" is het teken £ gebruikt !

Door bij de vraag naar de QTH-locator van het "tegenstation" op regel 80 als antwoord "return" te geven, wordt het programma beëindigd.
Bij mijzelf springt de computer terug naar "menu", omdat dit programma deel uitmaakt van een totaalprogramma met Morse- en RTTY-decodering.

J.H. Schoon.

<<<<<<<<<<< LISP 2 >>>>>>>>>>>

Nog even herhalen, een LISP Programma is een lijst. Een lijst is:

1: Een eventueel leeg woord of
2: Een lijst van lijsten.

Als een lijst niet een enkel wood is, dan staan de elementen vandie lijst tussen haakjes.
Bijvoorbeeld:

```
( EEN TWEE DRIE ) of
( VIER (( VIJF ZES ))) etc.
```

Het eerste element van een lijst is de FUNCTIENAAM en de eventuele andere elementen zijn de argumenten van de functie die bij die naam behoren. Alle LISP INTERPRETATOREN beschikken over een redelijk groot aantal standaard functies. Met behulp van deze standaard functies kan de gebruiker zijn eigen functies samenstellen en zo een LISP programma schrijven. Dit klinkt misschien ingewikkelt of omslachtig, maar in BASIC gebeurt het precies hetzelfde, alleen wordt er dan niet gepraat over functies, maar over statements. Zo wordt bijvoorbeeld:

```
A=123
```

in BASIC een "assignment" statement of "toekennings opdracht" genoemd, maar met hetzelfde recht kan je praten in LISP over een functie bijvoorbeeld "GEEFWAARDE" die twee argumenten heeft. Hier zijn dat "A" en "123". De waarde van de functie zelf is niet van belang.

De eerste LISP INTERPRETATOREN beschikten maar over een klein aantal standaard (of "primitieve") funties. Dit is echter geen nadeel, daar alle functies die men maar kan verzinnen uit deze standaard functies samengesteld kunnen worden. John McCarthy, de ontwerper van LISP heeft m.b.v. LISP ook bewezen, dat alle "effectief berekenbare" functies niets anders waren dan een samenstelsel van een klein aantal simpele functies.

De belangrijkste standaard functies van LISP zijn:

--- QUOTE ---

Dit is een simpele functie met een (1) argument, de waarde van de functie IS het argument. Toets maar eens in:

```
(QUOTE HALLO )
```

het antwoord is:

```
HALLO
```

Dit lijkt misschien eigenaardig of nutteloos, maar bedenk, dat LISP altijd

van woorden de WAARDE probeert uit te rekenen ("evalueren"). Als je in had getoets:

```
HALLO
```

dan had LISP de WAARDE van HALLO geprobeerd te vinden, en dat was 'm waarschijnlijk niet zo best gelukt. Misschien verduidelijkt het volgende voorbeeld iets.
Als je in BASIC intoets:

```
A=X
```

dan krijgt A niet de waarde 'X' (de letter X), maar de WAARDE van de letter X (of variabele), en dat is bij BASIC dan altijd een getal. Wil je in BASIC iets LETTERLIJK hebben, dan zet je het tussen QUOTES:

```
$A="HALLO"
```

Nu heeft A de waarde HALLO, en niet de waarde van HALLO, wat dat dan ook zijn mag.

Twee andere belangrijke standaard functies zijn:

--- CAR & CDR ---

De betekenis van de namen van deze functies is niet zo gemakkelijk te achterhalen zoals bijvoorbeeld met 'PLUS' of 'REMAINDER'.
De namen stammen uit de grijze oudheid van de IBM 7400 computer, waar de eerste LISP INTERPRETATOREN op gemaakt zijn. Deze computers hadden 30 BITS woordlengte, waarvan de eerste 15 BITS het ADRES gedeelte en de laatste 15 BITS het DATA gedeelte voorstelden van machinecode instructies.
Vandaar de namen Contents of Adress Part of Register en Contents of Data Part of Register. Maar nu genoeg geschiedenis. Deze twee functies hebben allebei een (1) argument, en dat moet een lijst zijn. Deze lijst mag niet leeg zijn en niet gelijk zijn aan een enkel woord. De waarde van de functie CAR is gelijk aan het eerste element van die lijst, en de waarde van CDR is gelijk aan de lijst uitgezonderd het eerste element. Bijvoorbeeld:

```
( CAR ( QUOTE ( DIT IS EEN LIJST )))
```

heeft als waarde DIT. Let op het gebruik van QUOTE!!
Het argument van CAR is:

```
( QUOTE ( DIT IS EEN LIJST ))
```

en LISP gaat dat evalueren. Het resultaat is:

```
( DIT IS EEN LIJST )
```

(zie de functie QUOTE) en hiervan wordt het eerste element genomen, dus: DIT.

Hadden we ingetypt:

```
( CAR ( DIT IS EEN LIJST ))
```

dan ziet LISP het woordje 'DIT' weer als een functie met als argumenten

IS,EEN en LIJST, en dan was alles in de war gelopen waarschijnlijk.

Bij CDR gebeurt er dit:

```
( CDR ( QUOTE ( DIT IS EEN LIJST )))
```

De waarde is nu:

```
( IS EEN LIJST )
```

De functies mogen ook door elkaar gebruikt worden:

```
( CAR ( CDR ( QUOTE ( DIT IS EEN LIJST ))))
```

Dit heeft als waarde: IS.
Om veel typwerk te vermijden zijn in ATOMLISP de functies CAR,CDR,CAAR,CADR,CDAR,CDDR opgenomen.
Deze laatste vier functies vervangen:

```
( CAR ( CAR .....
( CAR ( CDR .....
( CDR ( CAR .....
( CDR ( CDR .....
```

CAR en CDR splitsen een lijst in tweeen, de volgende functie knoopt ze weer aan elkaar:

--- CONS ---

Dit is een functie met twee argumenten, De waarde van CONStruct is een lijst gelijk aan het tweede argument met ervoor het eerste argument geplakt. Bijvoorbeeld:

```
( CONS ( QUOTE DIT ) ( QUOTE ( IS EEN LIJST )))
```

heeft als waarde:

```
( DIT IS EEN LIJST )
```

In ATOMLISP mag je om nog meer typwerk te vermijden inplaats van:

(QUOTE ZOMAARIETS) ook wel
' ZOMAARIETS typen, dat scheelt ook weer wat haakjes.

Stel dat LIJST een normale niet lege lijst is, en ook niet een enkel woord, dan geldt er voor CAR, CDR en CONS dat:

```
( CONS ( CAR LIJST ) ( CDR LIJST )) = LIJST
```

Let op bij CONS, dat het antwoord bij:

```
( CONS '( DIT IS ) '( EEN LIJST))
```

niet

```
( DIT IS EEN LIJST )
```

maar

```
(( DIT IS ) EEN LIJST )
```

is!! Het eerste argument wordt TOEGEVOEGD als nieuw eerste element van de tweede lijst!!

Nog een voorbeeldje:

```
(CONS(CADR LIJST)(CONS(CAR LIJST)(CDDR LIJST)))
```

heeft als waarde:

```
( IS DIT EEN LIJST )
```

Het woord 'LIJST' heeft de waarde (DIT IS EEN LIJST).

Dit laatste voorbeeldje was al bijna niet meer te lezen door de haakjes. Er is dan ook een nette manier gevonden om LISP programma's leesbaar op te schrijven in de volgende vorm:

```
( FUNCTIENAAM ARGUMENT1
              ARGUMENT2
                  .
                  .
              ARGUMENTn)
```

Dus het laatste voorbeeld had er dan zo uitgezien:

```
( CONS ( CADR LIJST )
       ( CONS ( CAR LIJST )
              ( CDDR LIJST )))
```

Dit is al heel wat leesbaarder. In ATOMLISP is een functie opgenomen, een zogenaamde "PRETTY PRINTER" of "SUPER PRINTER" die dat ook doet. Z'n naam is SPRINT. Dat vergemakkelijkt alles en beetje, en vermijdt een hoop getel van al die haakjes.

Volgende keer meer standaard functies en wat hopelijk minder saaie voorbeelden.
De standaardfuncties van de volgende keer bevatten onder andere de LISP PREDICATEN. In BASIC zijn dat de "TESTABLE EXPRESSIONS" (zie ATOM handleiding).

van

"Manual" »»»»»»» "Handboek"

evert sanders

deel 2

1.3.3. DELETE (=uitwissen of doorhalen)

Het voordeel van een tv scherm boven een stuk papier is dat fouten kunnen worden gecorrigeerd zonder dat er later iets van te zien is. De DELETE toets zorgt ervoor dat de laatste karakter(s) van de programmaregel waar je aan bezig bent worden uitgewist, de cursor gaat dus bij het indrukken van de DELETE toets een plaats terug. Om meerdere karakters snel uit te wissen kun je tegelijk met de DELETE toets de REPT toets indrukken. Het corrigeren gaat dan veel sneller. Vergissingen kunnen dus heel gemakkelijk en snel worden uitgewist.

LET OP : Als je na een programmaregel de RETURN toets hebt ingedrukt, kun je niet meer corrigeren met de DELETE toets. Tenminste niet meer in je vorige (reeds met RETURN afgesloten) programmaregel. Dus: corrigeren met de DELETE toets gaat alleen in de regel die nog niet is afgesloten met een RETURN. Hoe je in een vorige regel nog kunt corrigeren zullen we bespreken bij behandeling van de COPIE toets.

1.3.4. RETURN

De RETURN toets is een signaal voor de computer dat je klaar bent met het intypen van een regel met karakters. De cursor gaat na een RETURN naar het begin van de volgende regel, en de computer zal reageren op wat je ingetypt hebt door een eventueel antwoord uit te printen. (Of een ERROR geven).
Als je echter een programma aan het intypen bent zal de cursor na een RETURN alleen naar de volgende regel springen. Hij reageert dan natuurlijk niet op wat je hebt ingetypt. Inwendig reageert de computer natuurlijk wel: hij slaat de ingetypte regel in zijn geheugen op, zodat hij ten alle tijden weet wat er in zijn geheugen staat.

1.3.5. REPEAT - REPT

Als je de "repeat" toets, gemerkt met REPT, samen met een andere toets, bv de A, hebt ingedrukt, wordt de betreffende karakter, in dit geval dus de A steeds opniew op het scherm geschreven, totdat je de REPT toets loslaat. Of totdat de regel met 64 karakters is gevuld.
REPT is zeer nuttig in samenwerking met de DELETE toets om meerdere karakters zeer snel uit te wissen.
Bedenk echter dat als je alleen op de REPT toets drukt dit totaal geen effect heeft.

1.3.6. Control = CTRL

Er zijn verscheidene speciale functies beschikbaar via het toetsenbord welke worden verkregen door bepaalde toetsen samen met de "control" toets, gemerkt met CTRL, in te drukken. Voorlopig worden hier slechts de volgende twee controle functie's genoemd:
CTRL - G (dus de CTRL toets samen met de G toets indrukken) geeft een pieptoon in ATOM's luidspreker.
ZIE OOK 1.6 CTRL - L maakt het scherm schoon en zet de cursor links boven bij het begin.
LET OP: om na een CTRL - L verder te kunnen gaan moet je eerst op ESC drukken zodat de prompt weer verschijnt.
Doe je dat niet dan krijg je als je bv. LIST intypt een ERROR 94, dus een fout melding.
De "control" functie kunnen we ook in een programma verwerken. Hoe, dat zullen we later nog zien.

1.3.7. BREAK

De BREAK toets zal de computer resetten en hem terugbrengen in de toestand zoals die was nadat hij net aangezet is. Het moet normaal niet nodig zijn om de BREAK toets te gebruiken, maar sommige assembler programma's (wat dit zijn zal later nog worden besproken) kunnen een zgn "loop" veroorzaken, zie hoofdstuk 5, die op geen enkele andere mannier gestopt kunnen worden dan met de BREAK toets.
De inhoud van het geheugen wordt echter bewaart als de BREAK toets is ingedrukt geworden. Ieder opgeslagen programma kan weer worden teruggekregen.
Als je een BREAK hebt gedaan, typ dan OLD, geef een RETURN en typ dan LIST en zie daar is je programma weer. (LIST typen na OLD is echter niet noodzakelijk, je kan ook typen bv. OLD, RETURN geven en dan RUN).

1.4. SCROLLING

Als de cursor de onderkant van het scherm bereikt en je typt meer regels dan dat er op het scherm kunnen (er kunnen max. 16 regels op) dan gaat het scherm (eigenlijk de bovenste regel) "scrollen", dwz iedere regel schuift steeds een naar boven, zodat je altijd de laatste 16 regels ziet van wat je ingetypt hebt. De bovenste regel van de text op het scherm zal dus steeds verdwijnen.

1.5. Storing Text (bewaren of opslaan van gegevens , programma's cq text in het geheugen)

Iedere regel die achter ATOM's ' ' prompt wordt ingetypt en begint met een regelnummer wordt na een return niet meteen uitgevoerd, maar opgeslagen als tekst in het geheugen. Alle soorten inputs, data's enz. kunnen op deze manier worden opgeslagen.
Het zou bv de tekst van een boek, 'n basic programma, 'n assembler programma of data (=gegevens) voor een programma kunnen zijn.
Nu zullen we zien hoe je een stuk tekst of een programma moet intoetsen. Deze kan dan bv bewaart worden op cassette, of naar een printer gestuurd worden. Ook kunnen we de ingebrachte data cq tekst "editten" dwz bewerken, verbeteren, toevoegen ed.
Het navolgende is ook van toepassing als een nieuw programma moet worden ingetypt.
TIP: Wen je aan om voor een nieuw programma altijd eerst NEW in te typen, waarna een RETURN.

De regels van een programma moeten altijd beginnen met een regel- of lijnnummer. Dit mag ieder nummer zijn tussen 1 en 32767. Het hoogste regelnummer mag dan ook niet groter zijn dan 32767. Je hoeft de regelnummers ook niet achterelkaar te laten volgen, dus bv 1,2,3,4.
Dit is zelfs af te raden, je kunt beter de lijnnummers per bv 10 laten verspringen, dus bv 10,20,30,40. Waarom......?
dat zul je nog gau genoeg zien.
Achter het lijnnummer typ je dan je regel of tekst. Bv. typ het volgende:

```
10 VEEL COMPUTERS HEBBEN EEN VIDEOUITGANG
20 DEZE AANSL. GEEFT EEN VEEL SCHERPER BEELD
30  ALLE MONITORS HEBBEN EEN VIDEO- OF MONITORAANSLUITING
40 DE VIDEOAANSL. VAN UW TV IS HIERVOOR NIET GESCHIKT
```

Vergeet niet achter iedere regel een RETURN te geven. Met iedere regel bedoel ik een programmaregel, dus geen RETURN geven als de regel op het schem vol is maar pas als de programmaregel af is. Als de beeldschermregel vol is gewoon doortypen, het komt vanzelf op de volgende regel.
Iedere programmaregel kan 64 karakters (incl. spatie's, regelnummers, leestekens ed.) bevatten. Als je probeert meer dan 64 karakters in te typen, dan weigert ATOM verder te gaan. Je kunt dan ook geen RETURN geven om naar de volgende regel te gaan. Je zult dan eerst de laaste karakter met de DELETE toets moeten verwijderen, pas daarna kun je weer een RETURN geven en naar de volgemde regel gaan. Je kunt natuurlijk ook meerdere karakters verwijderen.
De reden waarom we de lijnnummers beter met bv 10 kunnen laten verspringen is dat we nu op een simpele manier regels kunnen tussenvoegen. Bv als een bepaalde regel te lang zou zijn.
Bv: om tussen bv regel 30 en 40 (zie vorig programmaatje) een regel tussen te voegen typ je gewoon achter de laatste regel (regel 40 in ons geval):

```
36 OOK ONZE COMPUTER HEEFT EEN VIDEOAANSLUITING
```

Typ nu LIST en zie ATOM heeft regel 36 reeds ertussen gesorteerd, en zelfs op de juiste plaats. Je zou een programma dus desnoods geheel door elkaar kunnen gooien (tenminste wat regelnummers betreft), ATOM zet de boel dan wel weer netjes op nummer.

1.6. COMMANDS (kommando's)

Commands die achter de '>' prompt worden getypt zonder dat ze voorafgegaan zijn door een regelnummer en gevolgt door een RETURN worden onmiddellijk door de computer uitgevoerd zonder dat ze eerst in het geheugen worden opgeslagen. Vandaar dat je deze dingen ook niet weer kunt opvragen.
Wil je ze wel opslaan in het geheugen dan MOET je lijnnummers gebruiken. Typ nu het kommando (de 5 regels uit het vorige voorbeeld moeten nu wel in het geheugen zitten):

LIST (=laat zien of print uit)

Dit kommando zal ervoor zorgen dat de opgeslagen text op het scherm geprint wordt.

Er zijn verschillende mogelijkheden met het LIST command bv.:

LIST	het hele progr. wordt tot het eind gelist
LIST 10	list alleen regel 10
LIST 20,40	list de regels 20 tm 40
LIST 20,	list de regels vanaf regel 20 tot einde
LIST ,30	list de regels vanaf de eerste regel tot regel 30.

TIP: soms wil de de listing rustig bekijken omdat er bv een fout ergens zit, maar als de listing bv 100 regels lang is, zou je steeds bv LIST 20,30 moeten doen, om toch alles gedurende een langere tijd op het scherm te houden.
Nu, het kan gemakkelijker: Typ : PRINT $14 , return, en typ nu LIST. Er verschijnen nu maar 16 regels op het scherm (de eerste keer kunnen er meer verschijnen zodat je de eerste niet meer ziet). Als er meer zouden verschijnen typ dan ; op ESC drukken en typ weer LIST nu gaat het goed.
Als nu de eerste 16 regels op het scherm staan en je wilt verder gaan hoef je alleen maar op de spacebar te drukken voor de volgende 16 regels.
En dat gaat zo door tot het einde.
Vergeet echter niet om als je klaar bent het volgende typen: PRINT $15, anders blijft de computer steeds 16 regels schrijven en stopt dan. Je kunt ipv PRINT $15 ook BREAK intoetsen en dan OLD typen.
Als je zou vergeten PRINT $15 in te typen is dat meestal nog niet zo erg want de meeste programma's lopen bij RUN dan toch wel gewoon door.

PRINT $14 is identiek aan CTRL N. De letter N is de veertiende letter van het alfabet, etc.

1.7 Editing (verbeteren, tussen voegen, weglaten ed.)

Een opvallend iets van de ATOM is dat opgeslagen text en/of programmaregels gemakkelijk kunnen worden gewijzigd door hetzelfde regelnummer (van de regel waar de fout inzit) te typen gevolgd door een nieuwe of verbeterde regel. Om bv regel 20 in een programma te veranderen typ:

```
20 DEZE REGEL IS VERANDERD
```

List het programma en zie het effect.
In de volgende aflevering gaan we verder in op het feit dat ATOM ook letters in een bestaande regel kan toevoegen of zelfs weghalen zonder dat er een spatie ontstaat.

Ter afsluiting volgt hier nog een klein programma:

```
10 P.$12
100 CLEAR4;Z=6;S=1+A.R.%4
110 F.Y=ØTO191S.S;MOVEØ,Y
120 PLOTZ,127,Ø;PLOTZ,255,Y;N.
130 S=1+A.R.%4
140 F.X=ØTO127S.S;MOVEX,191
150 PLOTZ,127,Ø;PLOTZ,(255-X),191;N.
160 LI.#FFE3;RUN
```

LET OP: het woord RUN in regel 160 moet via het toetsbord ingetypt worden

Als het programma af is (dus gerund) hoef je alleen maar op een willekeurige toets te drukken om opnieuw te starten.

overgenomen uit: ACORN MAGIC BOOK

AUTOMATISCHE REGELNUMMERING

Sinds ik de Acorn Atom heb, irriteer ik me er telkens aan dat ik bij het intikken van programma's zelf de nummering moet bijhouden en ingeven. Dat kan de computer zelf veel beter. Het eenvoudige routinetje, dat ik daarvoor maakte laadt men in een stuk geheugen met hoog adres,bijv. /8200. (Pointer op adres 18 aanpassen, "NEW" ingeven,en laden.) Na "RUN" vraagt de routine om een tekst-gebied. Geef in de eerste 2 posities van het adres, dus "/28" voor /2800, "/29" voor /2900, enz.
Daarna wordt gevraagd om het eerste regelnummer en de stappen voor verhoging. Bovendien wordt standaard in de eerste regel kommentaar geplaatst; de routine vraagt om een "Programmanaam".
Het "END"-statement hoeft U niet zelf meer in te tikken. Heeft U alle statements ingetikt, dan is een simpele Return genoeg om het programma te stoppen. De routine genereert een "END", en zet de pointer op adres 18 op het tekstgebied dat U zojuist gevuld hebt.Tik zelf nog even "END" om de "TOP"-pointer aan te passen.

```
10@=0;P.$3,$#C;IN."TEKSTGEBIED"A;?18=A;A=A*256;IN."STARTNR."B
50IN."STAPPEN VAN"C;?A=#D;A?1=B/256;A?2=B;B=B+C;A=A+3
100$A=" REM ";A=A+5;IN."PROGRAMMANAAM: "$A;DO;A=A+LEN(A)
150P.B;A?1=B/256;A?2=B;A?3=#20;A=A+4;IN.$A;B=B+C
210U.LEN(A)=0;$A="END";A=A+3;A?1=#FF;P."PROGRAMMA AFGELOPEN"
300E.
```

J.H. Man.

+_

"FIND" (string zoeken)

```
10 DIMA64; @=5
15AT=#29 ( T aanpassen aan eventuele andere
          geheugenruimte )
20 T=T*256+3
30 In.' "Welke string zoeken"$A;Z=LENA;I FZ<1 E.
40 DO W=LENT-1
50 FOR I=Ø to W;N=Ø
60CIF T? (I+N)<>A?N G.B
70 N=N+1; I FN=LENA;G=256x(T?-2)+T?-1;P. 'G,
   $T; I=W;G.B
80 G.C
90BN.I
100 T=T+LENT+3;U.T?-2=255
110 $.A
```

+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+

TOOLBOX aanzetten in een programma

```
2   !#20A=#F96EA141
3   !#202=#AØØØA1E2
9   X=Ø;Y=Ø
10  DATA 12,13
15  RESTORE
20  READ X,Y
30  !P.X,Y
35  BEEP 12,23
40  END
50  AFZETTEN TOOLBOX IN EEN
    PROGRAMMA MET
55  PROBEER MAAR EENS G.65;
60  !#2Ø2=#AØØØC9D8
70  !#2ØA=#F96EFE94
80  G.1Ø
```

A. Marchal.

+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+

NOISE VRIJ-tekenen

uit Byte

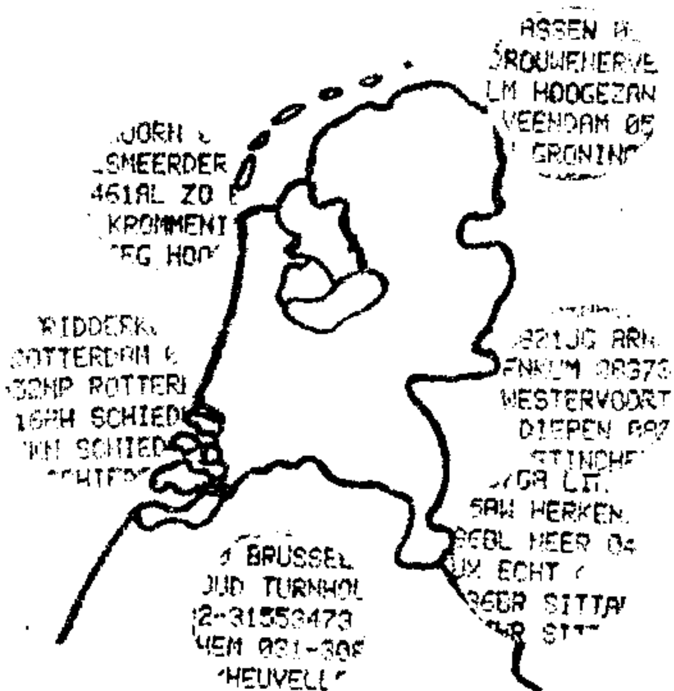
Na CLEAR X in het programma intoetsen :

```
?#80=?#3FE;?#81=?#3FF
?#3FE=TOP&#FF;?#3FF=TOP/256
A=TOP
!A=#22C80A9
A!4=#6CFBDOBO
?18=#80
```

Het tekenen van punten en trekken van lijnen gebeurt nu zonder interferentie op het scherm, maar gaat nu trager. Vertraging per punt: 1/60 sec.

Ingezonden door: A. Reynen.

U I T D E R E G I O's



Jawel : Uit de REGIO's. Je houdt het niet voor mogelijk maar de Regio's zijn in aantocht. En zoals dat zovaak gebeurt: als luie mensen vlot worden is er meteen geen houdan meer aan !

Bericht van REGIOVORMING:

+ Groningen/Friesland/Drenthe: Coördinator: Fens de Vries te Assen
+ Twenthe: Coördinator: Jan Jager te Wierden.
+ Arnhem/Achterhoek: Coördinator: Joop Uges te Ede.
+ Zuid Limburg: Coördinator: wordt nog overlegd. Voor contacten:fam Ernst te Echt.04754-5653.
+ Noord Holland: Coördinator: Jan Post te Aalsmeer--derbrug.
+ Rotterdam en omstr.: Coördinator: Gerard Akkermans te Rotterdam.
+ Zeeland: Coördinator: de Hr Heringa te 'sGraven--polder.
+ Belgie: Coördinator: Jaques Mijngheer te Brussel.

Alle genoemde Regio's hebben in de maand September een bijeenkomst gepland. Een kort verslagje hiervan zie ik graag tegemoet voor plaatsing in A.N.

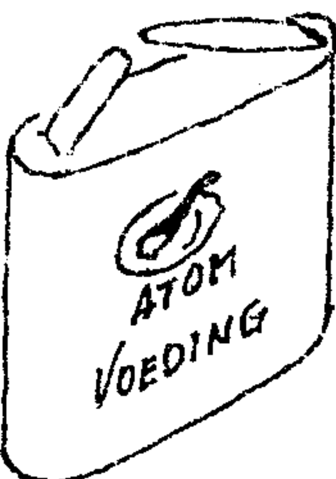
De gedachte die voorstaat bij REGIO-VORMING is: De Acorn Computerclub Nederland om te vormen tot een FEDERATIE van zo zelfstandig mogelijke regionale Clubs/Verenigingen. Uiteraard heb ik daarbij wel ideën hoe dit op de efficientste wijze zou zijn te realiseren. Ik ben echter bepaald niet van plan om die ideën erdoor te drukken. Eigenlijk integendeel. De gedachte is om b.v. in de maand October de Regionale Coördinatoren bijeen te roepen en aan deze vergadering de beslissingsbevoegdheid te laten over de vormgeving van de Vereniging. Uit deze vergadering kan dan de voordracht ontleend worden voor het eerste gekozen bestuur van de Acorn Club Nederland (Benelux ?). Deze voordracht wordt daaraanvolgend voorgelegd aan de eerste Algemene Vergadering; z.m. in November.

Aanbevolen wordt om per REGIO een paar functies te verdelen en aan mij door te geven, zoals:

- De ledenadministratie: We stormen op naar de 400 leden. Na de lijst in A.N. nr. 6 worden geen complete ledenlijsten meer gepubliceerd. Nieuwe leden worden aan de regio's doorgegeven.
- Een 'Software' contactman: Voor breder overleg over de 'soft' ontwikkeling van de Atom, verruiming van schakelmogelijkheden en het 'Upgraden'
- Een regionale ruilbeurs waarheen materiaal kan worden gezonden waarvan de inzenders enig voorbehoud maakten tov het (of hun) auteursrecht. Gebleken is, dat via onze 'officiele' wegen (AcornNieuws,Bandjesarchief) tóch eigen materiaal in de handel terchtkomt.
- Een regionale 'programmeerdienst'; bijzonder makkelijk voor het snel en goedkoop verspreiden van Utility-Eprom's zoals de 4k. UitbreidingsRom van Jos Horemmier. Elke binnen een Regio georganiseerde Epromdienst ontvangt van mij deze Eprom plus een T.U.V. (Wisser) lampje 5 Watt, aan ons cadeau gedaan door ons Belgische lid Roger Ennekens ! Echter ook hier gaat het om een vertrouwelijke functie. Jos geeft deze Eprom voor niets aan de club en vertrouwt erop dat dit zo blijft.(De Eprom wordt buiten de club door de E.C.D. verhandeld voor ƒ 89.-).

Borghaerts.

D E T O E K O M S T I S A L B E G O N N E N !



In vorig ACORN NIEUWS hadden we het over de toekomst waarin de CMOS IC's het stroomverbruik omlaag zouden brengen tot drukknop-cel niveau. Ik heb nog geaarzeld of ik dat wel zo zou stellen; waarom de profeet uithangen.

Welnu; die toekomst duurde nog geen 14 dagen. Via een tip van een van onze leden werden mij CMOS-varkens geleverd (½k) die voldoende gebufferd zijn om te kunnen solderen en stapelen.

Met deze CMOS-varkens werd een ATOM volgeprikt; gewoon de 2114 eruit en CMOS erin.

Het totale stroomverbruik van de ATOM, gortvol geheugen plus de schakelkaart daalde daardoor tot 0,7 Ampère (naar boven afgerond).

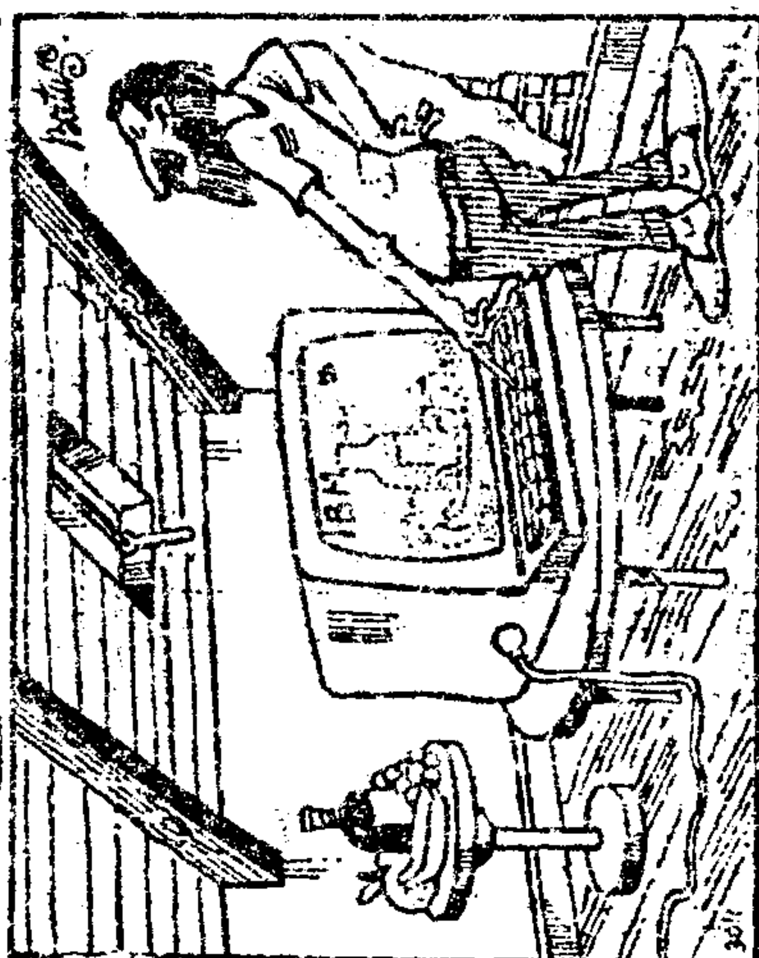
Per méér dan 1000 stuks komen deze CMOS-varkens op nog geen ƒ 8.- per stuk. Voor een 2114 wordt vaak méér gevraagd. (lp versie).

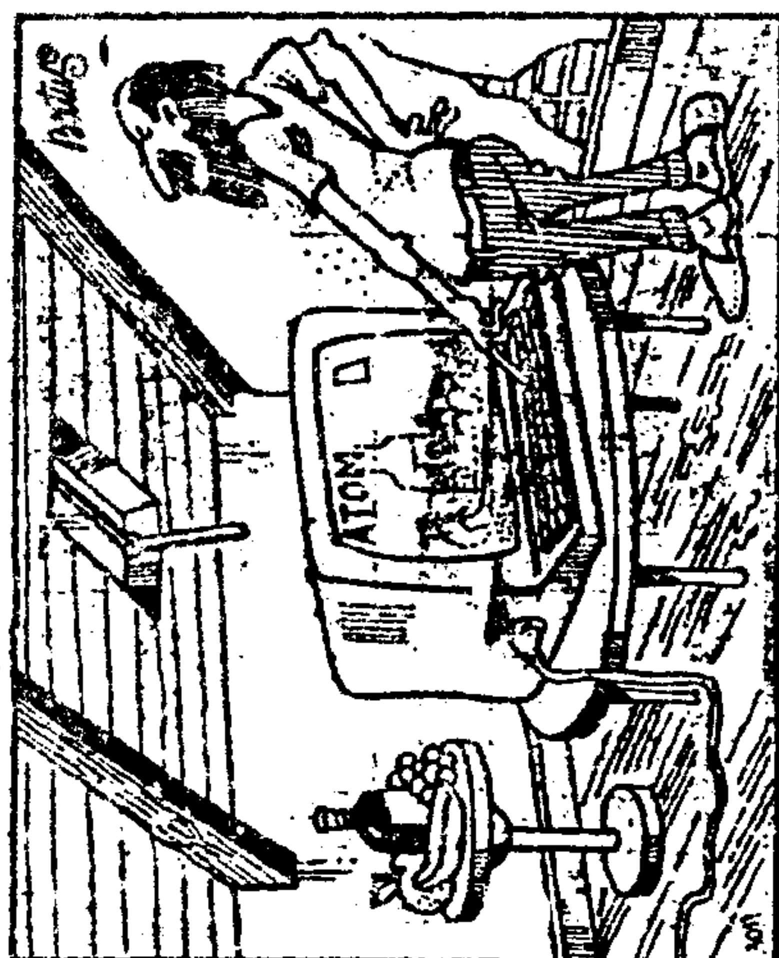
Ik ga U zeker niet voorstellen allemaal maar Uw geheugen IC's om te wisselen. Zou een kostbare grap worden. Wél is het zo, dat als een van U voor de noodzaak staat om een groter voedingsapparaat aan te schaffen, hij die ƒ 130.- beter kan besteden,door zijn varkens om te wisselen voor CMOS. Dat is vele malen efficienter. De uitgetrokken 2114's zouden zijn te bewaren om daar straks het onafhankelijke VIDEO-Geheugen van te maken, wég van ≠ 8000.

Wie belangstelling heeft laat dit het beste weten aan zijn Regionale Opperhoofd. Ik krijg dit dan wel per Regio door en dan zien we wel hoever we komen voor 'n gecombineerde inkoop.

= = = = = = = = = = = = = = = =

Zoek de tien kleine verschillen tussen de bovenste en onderste tekening





ACORN USER

Number 1 July/August 1982 £1.0

Acorn User
MAGSUB (Subscription Services) Ltd
Ground Floor Post Room
Oakfield House
Perrymount Road
HAYWARDS HEATH
West Sussex
RH16 3DH

